4

МОДЕЛИРОВАНИЕ И АНАЛИЗ ДАННЫХ

НАУЧНЫЙ ЖУРНАЛ

MODELLING AND DATA ANALYSIS

SCIENTIFIC JOURNAL

2020

ISSN: 2219-3758
ISSN: 2311-9454 (ONLINE)

МОДЕЛИРОВАНИЕ И АНАЛИЗ ДАННЫХ

НАУЧНЫЙ ЖУРНАЛ

2020 • Том. 10 • № 4

MODELLING AND DATA ANALYSIS

SCIENTIFIC JOURNAL

2020 • Vol. 10 • № 4

Московский государственный
психолого-педагогический университет
Moscow State University
of Psychology & Education

Издаётся с 2011 года

Учредитель
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский государственный психолого-педагогический университет»

Адрес редколлегии:
г. Москва, ул. Сретенка, 29, факультет информационных технологий
Тел.: +7 (499) 167-66-74
E-mail: mad.mgppu@gmail.com

Журнал зарегистрирован в Государственном комитете РФ по печати.
Свидетельство о регистрации средств массовой информации
ПИ № ФС77-52058 от 7 декабря 2012 года

ISSN: 2219-3758
ISSN: 2311-9454 (online)

СОДЕРЖАНИЕ

АНАЛИЗ ДАННЫХ

КОМПЛЕКСЫ ПРОГРАММ

МЕТОДЫ ОПТИМИЗАЦИИ

CONTENTS

Моделирование и анализ данных
2020. Том 10. № 4. С. 5–16
DOI: https://doi.org/10.17759/mda.2020100401
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2020. Vol 10, no. 4, pp. 5–16
DOI: https://doi.org/10.17759/mda.2020100401
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

АНАЛИЗ ДАННЫХ

УДК 004.855

Алгоритм машинного обучения для решения задачи формирования рекомендаций товаров и услуг

Судаков В.А.*
Московский авиационный институт (МАИ),
г. Москва, Российская Федерация
ИПМ им. М.В.Келдыша РАН, г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0002-1658-1941
e-mail: sudakov@ws-dss.com

Трофимов И.А.**
Московский авиационный институт (МАИ),
г. Москва, Российская Федерация
e-mail: trofimovc137@gmail.com

В статье предложен алгоритм машинного обучения без учителя для оценки наиболее возможных отношений между двумя элементами множеств клиентов и товаров/услуг с целью для построения рекомендательной системы. Рассмотрены методы на основе коллаборативной фильтрации и фильтрации на основе содержания. Разработан комбинированный алгоритм выявления отношений на множествах, сочетающий достоинства анализируемых подходов. Оценена сложность алгоритма. Даны рекомендации по эффективной реализации алгоритма с целью уменьшения объема используемой памяти. На примере задачи рекомендаций книг показано применение данного комбинированного алгоритма. Данный алгоритм может использоваться для «холодного старта» рекомендательной системы, когда ещё не существует размеченных качественных выборок обучения более сложных моделей.

Ключевые слова: машинное обучение, обучение без учителя, рекомендательные системы, сходство объектов, отношение, множество.

Для цитаты:
Судаков В.А., Трофимов И.А. Алгоритм машинного обучения для решения задачи формирования рекомендаций товаров и услуг // Моделирование и анализ данных. 2020. Том 10. № 4. С. 5–16. DOI: https://doi.org/10.17759/mda.2020100401

****Судаков Владимир Анатольевич,*** д.т.н., профессор каф.805, Московский авиационный институт (МАИ), ведущий научный сотрудник, Федеральное государственное учреждение «Федеральный исследовательский центр Институт прикладной математики им. М.В. Келдыша Российской академии наук» (ИПМ им. М.В.Келдыша РАН), ORCID: https://orcid.org/0000-0002-1658-1941, e-mail: sudakov@ws-dss.com

*****Трофимов Иван Александрович,*** студент, Московский авиационный институт (МАИ), e-mail: trofimovc137@gmail.com

1. ВВЕДЕНИЕ

В связи со стремительным распространением алгоритмов и методов машинного обучения в решении прикладных задач, на сегодняшний день появилось множество сфер их применения в бизнес-задачах повышения эффективности, получения прибыли, постановки прогнозов и так далее. Одним из направлений машинного обучения, которое сейчас особенно востребовано в прикладной области является сфера рекомендательных систем.

Увеличение роли рекомендательных моделей связано как с широким распространением интернет-продаж, а соответственно и интернет-рекламы, которая довольно адаптивна и может легко подстраиваться под конкретного пользователя, так и с наличием огромного количества информации, из которой можно тем или иным способом получить предпочтения клиента и сделать рекомендацию ориентированной под его специфические интересы и запросы [2].

Проблемой зачастую является качество самих исходных данных для построения прогноза. Эффективные модели требуют размеченных данных для обучения с учителем (например рейтинги, оставленные пользователями для каждого товара), либо наоборот, модели ориентируются лишь на схожесть товаров относительно друг друга, не учитывая опыт других пользователей.

В данной статье предлагается алгоритм построения ранжированных релевантных рекомендаций, основывающийся как на схожести пользователей, так и на сходстве товаров и услуг и оценивающий наиболее вероятные, но на данный момент ещё не появившиеся отношения между элементами двух исходных множеств.

2. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

На данный момент в рекомендательных системах существует два основных подхода создания рекомендаций – коллаборативная фильтрация и фильтрация на основе содержания [2].

Метод коллаборативной фильтрации использует данные об известных предпочтениях пользователей для прогнозирования предпочтений для других пользователей. Алгоритм основывается на допущениях о том, что пользователи, совместно оценившие схожим образом некоторую группу товаров, будут схожим образом оценивать и те товары, которые ещё не были оценены одним из пользователей [1]. Для наглядности данное предположение проиллюстрировано в таблице 1.

Таблица 1

Таблица коллаборативной фильтрации

	Товар № 1	**Товар № 2**	**...**	**Товар № N**
Пользователь № 1	5	4		4
Пользователь № 2	5			3
…				
Пользователь № M	?	4		4

Далее для активного прогнозируемого значения ячейки «пользователь – товар» формируется прогноз на основе информации о соседстве (фильтрация, основанная на соседстве) либо с помощью модели машинного обучения (фильтрация, основанная на модели). Прогнозируется значение – это значение, которое пользователь поставил-бы данному товару, если бы ему пришлось его оценивать.

Преимущества метода коллаборативной фильтрации:

- Метод является универсальным для данных подходящего формата.
- Для работы не требуется детального описания товара.
- В большинстве практических задач метод имеет высокую точность.

Недостатки метода коллаборативной фильтрации:

- Обучающие данные должны быть размечены (необходимо иметь оценки пользователей). Зачастую, при начале работы рекомендательной системы, нет данных о предпочтениях пользователей. Также имеющихся данных может быть недостаточно для оценки поведения пользователя и предсказания этого поведения (один из вариантов работы алгоритма) [2].
- Разреженность данных и масштабируемость. Многие пользователи могут не оставлять пользовательскую оценку (рейтинг) товара. Но даже если такие данные есть, то товаров, как правило огромное количество, как и пользователей, что приводит к тому, что обучающие данные крайне разрежены и обладают большой размерностью. Из этого вытекает как существенные затраты на вычисления (сложность алгоритма O(M*N)), так и проблемы, связанные с понятием «проклятие размерности», при котором новые данные могут лежать крайне далеко от обучающих образцов.
- Проблема холодного старта. Трудно рекомендовать товары, недавно добавленные в систему, или формировать рекомендации для пользователей, которые ещё ничего не выбирали [3].
- Синонимия. Схожие объекты с различными именами могут быть рассмотрены системой как различные. Например «книги для детей» и «детская литература».
- Существуют пользователи, чье мнение не совпадает с мнением большинства и для них трудно что-либо рекомендовать [2].

Другой метод – это метод фильтрации на основе содержания. Он основан на наличии описания о рекомендуемых объектах. Описание состоит из некоторого набора характеристик элемента. Значения характеристик должны в том или ином виде существовать для начала работы алгоритма [4].

Каждому пользователю ставится в соответствие профиль, содержащий характеристики, схожие с характеристиками рекомендуемых объектов. Как правило пользователя прямо спрашивают о его предпочтениях (например, просят указать наиболее интересные жанры), либо профиль формируется из истории уже заказанных пользователем товаров.

Преимущества метода фильтрации на основе содержания:

- Не требуется наличие большой группы пользователей для формирования рекомендации.
- Новые элементы могут быть рекомендованы сразу, после добавления их характеристик.

Недостатки метода фильтрации на основе содержания:

- Необходимо подробное описание товаров (большое количество характеристик), для возможности наиболее точно их различать.
- Метод зависит от предметной области и полезность полученных рекомендаций ограничена.
- Характеристики профиля пользователя должны быть сопоставимы с характеристиками предметов.
- Метод выдает рекомендации из «ближайших соседей» предмета, не рекомендуя что-то новое.
- Метод не учитывает опыт других пользователей.

Как видно, основные подходы в построении рекомендаций обладают рядом жестких требований к данным [4]. Необходимо либо наличие рейтингов пользователей о товаре, либо подробное описание товара в виде характеристик и наличие профиля пользователя.

Представленный в данной статье метод использует идеи из обоих предыдущих методов, при этом предоставляя гораздо более низкие требования к качеству данных для обучения. При этом отсутствия каких-либо данных приведет к тому, что метод просто вырождается в частный случай подхода, основанный на близости пользователей (если нет никакой информации о товаре), либо в частный случай фильтрации на основе содержания (если невозможно определить какую-либо информацию о сходстве пользователей) [5]. Но как будет показано в части с практическим применением метода, достаточно исторических данных о том, какие товары заказывал пользователь.

Также преимуществом метода является то, что полученным рекомендациям будет поставлен в соответствие численный рейтинг рекомендации, что позволит упорядочить рекомендации по релевантности. При этом для этого не требуются оценки товаров пользователями.

3. ПОСТАНОВКА ЗАДАЧИ И АЛГОРИТМ РЕШЕНИЯ

Пусть имеется множество пользователей $U = \{u_1, \ u_2, \ \ldots, \ u_n\}$, множество объектов $O = \{o_1, \ o_2, \ \ldots, \ o_m\}$ и множество отношений между пользователями и объектами $C = \{c_1, \ c_2, \ \ldots, \ c_x\}$, где $c_i = (u_j, \ o_k)$ – наличие отношения между пользователем u_j и объектом o_k. Требуется для пользователя u_s оценить ещё не существующие, но наиболее вероятные отношения с элементами множества O.

Описание алгоритма:

1. Оценка расстояний между всеми элементами множества U с помощью некоторой функции $Fu(u_1, \ u_2)$. Функция Fu будет оценивать «схожесть пользователей» относительно друг друга. Данная функция является одним из гиперпарамметров модели. При наличии численных признаков $Ux = \{ux_1, \ ux_2, \ \ldots, \ ux_n\}$ – это может быть метрика в многомерном пространстве, либо Гаусовская радиальная функция близости. При отсутствии каких-либо признаков пользователей, в качестве меры близости используются функции пересечения множеств используемых

товаров, например расстояние Жаккарда. Далее, в практическом примере, используется метрика на основании пересечения множеств.

2. Оценка расстояний между всеми объектам рекомендаций O. Аналогично расстоянию между пользователями, вводится некоторая функция близости объектов $Fo(o_1, \quad o_2)$. Данная функция является вторым гиперпараметром модели и выбирается в зависимости от поставленной задачи и пожеланий в результатах. В практическом примере в качестве такой функции использовалась косинусная схожесть векторизованных текстовых описаний объектов [6].
3. Оценивается численный показатель возможного отношения каждого объекта множества O с пользователем u_s, для которого строится прогноз, как взвешенная сумма существующих отношений, умноженных на коэффициенты близости пользователей и объектов, сопоставленных с этим отношением:

$$r_{oi}(u_s, \quad o_p) = \sum_{c_i \in C} Fu(u_s, \quad u_{ci}) \bullet Fo(o_{ci}, \quad o_p), \text{ где:}$$

$c_i = (u_{ci}, \quad o_{ci})$ – связь между пользователем u_{ci} и объектом o_{ci},
u_s - пользователь, для которого составляется рекомендация,
$Fu(u_s, \quad u_{ci})$ – близость пользователя из c_i с пользователем u_s, для которого вычисляется прогноз,
o_p - объект, для которого оценивается возможность существования связи с u_s,
$Fo(o_{ci}, \quad o_p)$ – близость анализируемого объекта o_p и объекта o_{ci},
u_s - пользователь, для которого составляется рекомендация.

Вычисление возможности связи выполняется для всех пар $(u_s, \quad o_p)$, где наличие этой связи ещё не подкреплено экспериментально: $(u_s, \quad o_p) \notin C$.

4. Для каждого пользователя u_s объекты o_p ранжируются по полученному для них рейтингу $r_{oi}(u_s, \quad o_p)$.

Данный алгоритм исходит из двух предположений об имеющихся данных:

- Пользователи образуют группы, в которых интересы близких пользователей совпадают.
- Объекты образуют группы, близкие объекты в которых объединены общей тематикой и интересуют схожих пользователей.

Для релевантных результатов алгоритма достаточно выполнения хотя-бы одного из этих предположений.

При отсутствии схожих групп пользователей, их связи будут учитываться с малым коэффициентом (из-за удаленности друг от друга). Либо не будут иметь пересечений между собой. В таком случае модель вырождается в рекомендации товаров, наиболее похожих на те, которые пользователь уже покупал.

При отсутствии схожести в группах товаров аналогичным образом товары, не имеющие связей с группой ближайших пользователей, будут иметь малые веса либо не будут пересекаться. В таком случае модель вырождается в рекомендации тех товаров, которые наиболее популярны у схожих пользователей (по количеству связей, умноженных на близость пользователей).

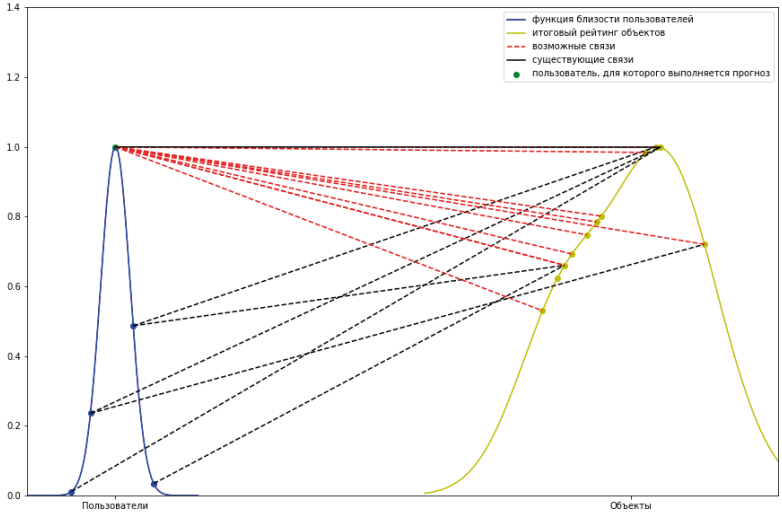

Рис. 1. Пример отношения между множествами

На рисунке 1. Приведен пример построения прогнозируемых отношений между множеством пользователей и множеством объектов. В данном примере у пользователей и объектов есть ряд численных признаков, а близость между объектами определяется Гаусовой радиальной функцией. Синим графиком отмечена «близость» прочих пользователей к тому, для которого строится рекомендация. Черными линиями отображены существующие связи между множествами. Желтый график – итоговый рейтинг каждого объекта для рекомендации пользователю. Красный пунктир – возможные рекомендации. Каждая рекомендация имеет свой рейтинг, по которому может проведено ранжирование.

4. СЛОЖНОСТЬ АЛГОРИТМА

Первый шаг алгоритма имеет сложность $O\left(\frac{M^2}{2}-M\right)$ для расчета схожести пользователей если Fu – симметричная функция, в противном случае $O\left(M^2-M\right)$. А второй шаг алгоритма имеет сложность $O\left(\frac{N^2}{2}-N\right)$ для расчета схожести товара.

При полном расчете приходится хранить матрицы соответствующего размера, что может потребовать больших затрат памяти. Для решения проблемы с памятью можно поступить следующим образом: для пользователей и объектов рассчитать схожести внутри случайно-взятых подмножествах выборки. Определить квантили q_u и q_o для метрики близости внутри объектов и учитывать только тех соседей, у которых степень «близости» выше данного порогового значения. Таким образом можно хранить лишь таблицу вида: «объект-множества»: «ближайшие соседи и их близость».

Это значительно сэкономит используемую память (даже по сравнению с разреженными массивами, т.к. большинство объектов имеют очень низкую, но не нулевую близость), а для формирования прогноза надо будет сделать лишь следующие действия:

- взять ближайших соседей пользователя,
- взять соответствующие их связям объекты,
- из таблицы объектов выбрать ближайших соседей к объектам,
- рассчитать ранги всех объектов как взвешенные суммы произведений функций близости.

В данном случае квантили представляют собой ещё два гиперпараметра модели, которые можно подобрать в зависимости от задачи.

5. ПРИМЕР ПРАКТИЧЕСКОГО ИСПОЛЬЗОВАНИЯ АЛГОРИТМА

Данный алгоритм был использован при решении задачи формирования рекомендации читателям библиотек по имеющимся данным о выдаче книг. Эта задача была представлена на хакатоне «Лидеры цифровой трансформации» от департамента культуры города Москвы. Задача включала в себя несколько разделов, в качестве примера приведены рекомендации по книгам.

Исходные данные содержали:

- историю пользователей в формате «id пользователя» и «id взятых книг»,
- каталог книг, содержащий краткую текстовую информацию о книге или журнале: название, автор, издательство, жанр, возрастные ограничения.

В данных отсутствуют профили пользователей, рейтинги или предпочтения пользователей, а также нет критериев того, что считается оптимальной рекомендацией. Также требованием является наличие в рекомендациях объектов, не похожих на то, что он читал ранее.

Данные особенности накладывают существенные ограничения на использование базовых алгоритмов рекомендательных систем. Отсутствуют рейтинги пользователей. Интерпретация взятия книги как рейтинга $\langle 1 \rangle / \langle 0 \rangle$ приводит к не релевантному результату рекомендации. Информации о объектах хватает лишь для разбиения на крупные множества.

Данные особенности делают задачу хорошим вариантом для тестирования предложенного метода.

Определение гиперпараметров модели:

1. В качестве функции близости пользователей была определена функция отношения пересечения множеств взятых книг ко множеству книг, взятых первым пользователем: Какую долю из книг первого пользователя уже брал второй. Данная функция не является симметричной. Функция используется для более сильной ориентации в прогнозе на пользователей, которые брали большее количество книг. В тоже время пользователь, имеющий небольшой «опыт» использования системы будет иметь меньшую схожесть.

2. В качестве функции близости между объектами (книгами) использовалась косинусная похожесть векторизированных описаний книг.
3. В модели не производился полный расчет таблиц, применялась оптимизация по памяти, описанная ранее. Квантиль для пользователей $q_u = 0.7$, квантиль для схожести книг $q_o = 0.8$.
4. После определения границы ближайших соседей, функции близости были упрощены до ступенчатых (1 если объект входит в границу).

Параметры были оценены при обучении на подвыборке из $N = 2000$ пользователей и $M = 5000$ книг.

Примеры рассмотрим примеры рекомендаций, выдаваемых моделью для разных пользователей.

Прочитанные первым случайным пользователем книги: автор – «Устинова Татьяна Витальевна», название – «Пять шагов по облакам», жанр – «Художественная литература»; автор – «Устинова Татьяна Витальевна», название – «Седьмое небо», жанр – «Художественная литература». Топ 5 рекомендаций, рассчитанных по модели пользователя, вошли книги, показанные на рисунке 2.

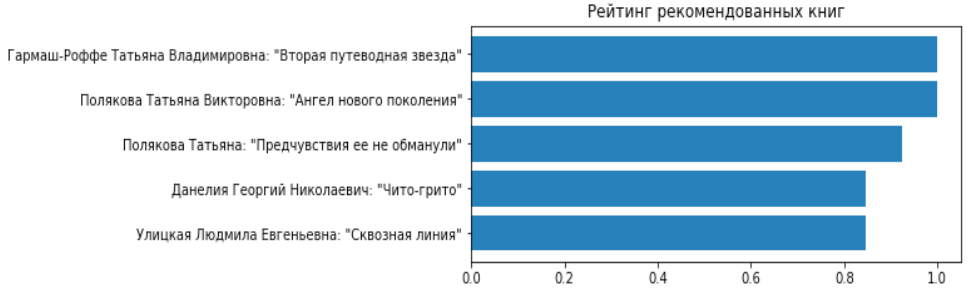

Рис. 2. Пример рекомендаций для первого случайного пользователя

Для пользователя, взявшего всего одну книгу: автор – «Тимм Уве», название – «Руди-Пятачок», жанр – «Художественная литература», топ 10 рекомендаций, рассчитанных по модели пользователя, вошли книги, показанные на рисунке 3.

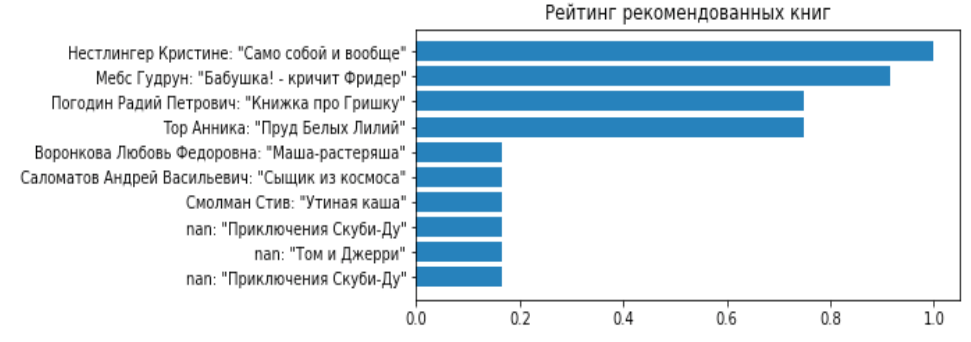

Рис. 3. Пример рекомендаций для второго случайного пользователя

Для пользователя с узкоспециализированным запросом были взяты книги: автор – «Пушкарева Наталья Львовна», название – «Частная жизнь русской женщины XVIII века», жанр – «Этнология (этнография, народоведение); автор – «Данелия Георгий Никола-евич», название – «Чито-грито», жанр: «Художественно-игровое кино». Топ 10 рекомендаций системы показан на рисунке 4.

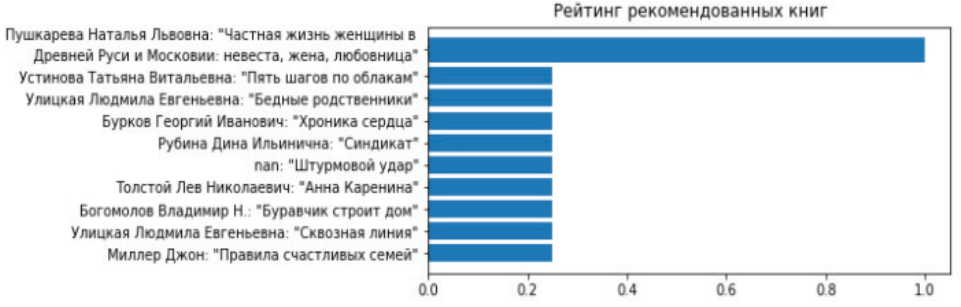

Рис. 4. Пример рекомендаций для пользователя с узкоспециализированным запросом

6. ЗАКЛЮЧЕНИЕ

Даже при существенном упрощении модели: ступенчатые функции близости, обучение лишь на малой части выборки, уже можно сказать, что модель показывает релевантные результаты рекомендаций. При необходимости данную модель можно настроить более тонким образом, используя более сложные функции близости и подобрав квантили, отвечающие за регуляризацию.

Преимущества модели:

- Низкие требования к качеству данных. Достаточно того, чтобы на одном из множеств было возможно определить функцию близости элементов множества.
- Обучение без учителя. Для модели не требуется размечать данные. Хватает лишь наличия отношений между множествами.
- Ранжированный результат. Модель выдает не только рекомендации, но и их ранг. При использовании сложных функций близости (таких например, как Гаусова радиальная функция RBF), полученные значения можно будет интерпретировать как уверенность в «удачной» рекомендации.
- Решена проблема холодного старта. Новый пользователь либо имеет связь с хотя-бы одним предметом, либо на множестве пользователей определена функция близости, не зависящая от «опыта» пользователя.
- За счет учета как опыта других пользователей, так и схожести объектов, генерируются разнообразные рекомендации.
- Частично решена проблема для «уникальных» пользователей, интересующихся узким кругом объектов. Рекомендации для них будут релевантны, если такие пользователи, либо объекты, интересующие их, формируют небольшую локальную группу. В таких случаях фактически рекомендации локализованы в подмно-

жестве основного множества объектов. Но при этом в рекомендации будут попадать и популярные для всех объекты, правда с меньшим рейтингом.

- Возможность оптимизации модели с помощью гиперпараметров.

Недостатки модели:

- Высокая вычислительная сложность, связанная с расчетом двух таблиц схожести.
- Низкая точность на пользователях, с большим «разбросом» интересов. Таким рекомендуются наиболее популярные объекты.
- Вычислительная сложность оперативного добавления новых элементов (необходимость считать схожесть с остальной выборкой), а в особенности при использовании несимметричных функций похожести (расчет схожести всей выборки с новым элементом)

Из совокупности имеющихся недостатков и преимуществ данного метода, можно сделать вывод, что модели подобного рода могут использоваться как модели «холодного старта» рекомендательной системы, т.е. являться «базовыми моделями» [2]. Когда ещё не существует размеченных качественных выборок для обучения более сложных моделей. На основе подобной модели можно составить первичные рекомендации, проверить их, оценить и сформировать выборку данных для более сложных моделей. Таких, например, как колаборативная фильтрация, в которой можно использовать полученные «базовой моделью» ранги [1].

Данный подход может использоваться и в других классах задач, не связанных с системами рекомендаций. Например, при анализе пользователей социальных сетей и сопоставления подмножеств групп пользователей друг с другом.

Литература

1. *Melville P., Mooney R., Nagarajan R.* Content-Boosted Collaborative Filtering for Improved Recommendations // University of Texas, USA : Материалы конф. / AAAI-02, Austin, TX, USA, 2002. – 2002. – pp. 187–192.
2. *Jannach D., Zanker M., Felfering A., Friedrich G.* Recommender Systems: An Introduction. Cambridge University Press, 2010.
3. *Ricci F., Rokach L., Shapira B., Kantor P.* Recommender Systems: Handbook. Springer, 2011.
4. *Linden G., Smith B., York J.,* Amazon.com recommendations: item-to-item collaborative filtering // Internet Computing – IEEE 7 2003 – pp. 76–80.
5. *Melville P., Mooney R.J., Nagarajan R.* Content-boosted collaborative filtering for improved recommendations // in Proceedings of the National Conference on Artificial Intelligence – 2002 – pp. 187–192.
6. *Белова К.М., Судаков В.А.* Исследование эффективности методов оценки релевантности текстов // Препринты ИПМ им. М.В. Келдыша 2020. No 68. 16 с. http://doi.org/10.20948/prepr-2020–68

The Machine Learning Algorithm for Solving the Problem of Generating Recommendations for Goods and Services

Vladimir A. Sudakov*
Moscow Aviation Institute (MAI), Moscow, Russian Federation
Keldysh Institute of Applied Mathematics (Russian Academy of Sciences), Moscow, Russian Federation
ORCID: https://orcid.org/0000-0002-1658-1941
e-mail: sudakov@ws-dss.com

Ivan A. Trofimov**
Moscow Aviation Institute (MAI), Moscow, Russian Federation
e-mail: trofimovc137@gmail.com

The article proposes an unsupervised machine learning algorithm for assessing the most possible relationship between two elements of a set of customers and goods / services in order to build a recommendation system. Methods based on collaborative filtering and content-based filtering are considered. A combined algorithm for identifying relationships on sets has been developed, which combines the advantages of the analyzed approaches. The complexity of the algorithm is estimated. Recommendations are given on the efficient implementation of the algorithm in order to reduce the amount of memory used. Using the book recommendation problem as an example, the application of this combined algorithm is shown. This algorithm can be used for a "cold start" of a recommender system, when there are no labeled quality samples of training more complex models.

Keywords: machine learning, unsupervised learning, recommender systems, object similarity, relation, set.

For citation:
Sudakov V.A., Trofimov I.A. The Machine Learning Algorithm for Solving the Problem of Generating Recommendations for Goods and Services. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2020. Vol. 10, no. 4, pp. 5–16. DOI: https://doi.org/10.17759/mda.2020100401 (In Russ., abstr. in Engl.).

*** ***Vladimir A. Sudakov,*** Doctor of Technical Sciences, Professor of Department 805, Moscow Aviation Institute (MAI), Leader Researcher, Keldysh Institute of Applied Mathematics (Russian Academy of Sciences), ORCID: https://orcid.org/0000-0002-1658-1941, e-mail: sudakov@ws-dss.com

**** ***Ivan A. Trofimov,*** student, Moscow Aviation Institute (MAI), e-mail: trofimovc137@gmail.com

References

1. Melville P., Mooney R., Nagarajan R. Content-Boosted Collaborative Filtering for Improved Recommendations. University of Texas, USA. *Proceeding of AAAI-02*, Austin, TX, USA, 2002. – 2002. – pp. 187–192.
2. Jannach D., Zanker M., Felfering A., Friedrich G., Recommender Systems: An Introduction. Cambridge University Press, 2010.
3. Ricci F., Rokach L., Shapira B., Kantor P. Recommender Systems: Handbook. Springer, 2011.
4. Linden G., Smith B., York J., Amazon.com recommendations: item-toitem collaborative filtering. *Internet Computing* – IEEE 7 2003 – pp. 76–80.
5. Melville P., Mooney R.J., Nagarajan R. Content-boosted collaborative filtering for improved recommendations. *Proceedings of the National Conference on Artificial Intelligence* – 2002 – pp. 187–192.
6. Belova K.M., Sudakov V.A. Issledovanie effektivnosti metodov ocenki relevantnosti tekstov [Research of the effectiveness of methods for assessing the relevance of texts]. *Preprinty IPM im. M.V. Keldysha = Keldysh Institute preprints*, 2020. No 68. 16 p. http://doi.org/10.20948/prepr-2020–68. (In Russ.).

Моделирование и анализ данных
2020. Том 10. № 4. С. 17–30
DOI: https://doi.org/10.17759/mda.2020100402
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2020. Vol 10, no. 4, pp. 17–30
DOI: https://doi.org/10.17759/mda.2020100402
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

УДК 004.85

Прогнозирование покупки товара, показанного клиенту рекомендательной системой

Парфенов П.А.*
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0001-5995-347X
e-mail: pentalbymf@mail.ru

Тимофеева А.А.**
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0001-7043-3715
e-mail: alena195101@yandex.ru

Сологуб Г.Б.***
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0002-5657-4826
e-mail: glebsologub@ya.ru

Алексейчук А.С.****
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация

Для цитаты:
Парфенов П.А., Тимофеева А.А., Сологуб Г.Б., Алексейчук А.С. Прогнозирование покупки товара, показанного клиенту рекомендательной системой // Моделирование и анализ данных. 2020. Том 10. № 4. С. 17–30. DOI: https://doi.org/10.17759/mda.2020100402

***Парфенов Павел Андреевич,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0001-5995-347X, e-mail: pentalbymf@mail.ru

****Тимофеева Алена А.,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация ORCID: https://orcid.org/0000-0001-7043-3715, e-mail: alena195101@yandex.ru

*****Сологуб Глеб Борисович,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0002-5657-4826, e-mail: glebsologub@ya.ru

******Алексейчук Андрей Сергеевич,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация

В работе рассматриваются различные методы улучшения рекомендательных систем. Проводится сравнительный анализ двух моделей для решения задач классификации: случайного леса (Random Forest) и CatBoostClassifier. Исследование выполнялось на данных истории покупок клиентов компании Ozon. Были использованы стандартные методы, часто применяемые в рекомендательных системах. Были реализованы методы коллаборативной фильтрации, косинусная схожесть товаров от просмотров клиента за одно посещение сайта, схожесть текстовых данных. Для оценки результатов использовались специальные метрики, оценивающие качество предсказаний первых k объектов из рекомендаций: Mean average precision (map@K) и Recall at K (recall@k). При генерации дополнительных признаков, основанных на различных методах, выявляющих схожесть объектов, отмечается увеличение качества прогнозов моделей. Модель CatBoostClassifier показала наилучшие результаты.

Ключевые слова: рекомендательные системы, машинное обучение, бинарная классификация, методы коллаборативной фильтрации, косинусная схожесть, map@K, recall@k.

1. ВВЕДЕНИЕ

В данной работе рассматриваются методы улучшения рекомендательных систем на примере компании Ozon, которая представляет собой один из ведущих интернет-магазинов на российском рынке. Для подобного интернет-магазина очень важно помогать покупателям находить необходимые товары быстрее, предоставляя возможность клиентам затрачивать наименьшее количество усилий и времени на поиски нужной вещи. С этой задачей отлично справляется рекомендательная система. Рекомендательная система – это система, определяющая рекомендации по предметам, которые могут быть полезны пользователю [1]. Так, когда человек смотрит или ищет что-то на сервисе, система выявляет то, что может ему понравиться и предлагает это, благодаря чему пользователь может быстрее найти необходимое, у него останутся хорошие впечатления о совершенной покупке, а магазин увеличит продажи за счет сбыта большего количества товаров. Одним из способов улучшения взаимодействия между сервисом, предоставляющим услуги, и клиентом, желающим ими воспользоваться, может быть улучшение рекомендательной системы, чтобы лучше удовлетворять его потребности.

В современном мире любая компания, которая предоставляет свои услуги через сеть интернет, сталкивается с проблемой выдачи релевантного контента своим пользователям. Рекомендация является релевантной, если она соответствует запросам пользователя [2]. Рекомендательные системы являются неотъемлемой частью продукта, через который происходит взаимодействие с будущими покупателями. Узнавая профиль клиента, его предпочтения и интересы, можно находить и выдавать для него необходимые ему услуги и товары. Большинство современных сервисов, предоставляющих свои услуги через интернет, используют системы рекомендаций. Например, компания Amazon. Она является одной из крупнейших платформ электронной коммерции, которая предоставляет множество услуг, от интернет-магазина до онлайн кинотеатра.

Amazon понимает, насколько важно правильно взаимодействовать с пользователем, и еще в 2000 годах компания реализовала свою первую рекомендательную систему и до сих пор продолжает работу по улучшению своих рекомендаций. Компания верит, что рекомендации должны сделать опыт взаимодействия с сервисом лучше [3].

В некоторых отраслях рекомендательные системы особенно важны, так как при эффективной настройке рекомендаций они могут помочь развитию бизнеса. Например, в онлайн-кинотеатре Netflix считают, что основой их бизнеса является именно рекомендательная система. Их рекомендательная система помогает компании в удержании клиентов: когда зритель начинает смотреть фильм, кинотеатр помогаем ему найти что-то интересное в течение нескольких секунд, предотвращая отказ от сервиса в пользу альтернативного варианта развлечения. Также благодаря рекомендациям определится подходящая категория пользователей для специфических фильмов, которые при попытке транслирования на традиционном телевидении принесли бы убытки, так как не нашли бы большого отклика среди зрителей и не смогли бы поддерживать значительный доход от рекламы [4]. Хорошо настроенная рекомендательная система позволяет удерживать старых пользователей и на основании их позитивного опыта привлекать новых. Поэтому развитие и улучшение рекомендаций очень важная задача, решение которой полезно как для бизнеса, так и для клиентов.

2. ПОСТАНОВКА ЗАДАЧИ

В данной работе рассматривается рекомендательная система компании Ozon. Когда покупатель заходит на сайт и страницу какого-нибудь товара, он видит описание товара, его цену и отзывы (рис.1). Внизу страницы товара есть графа «Рекомендуем также», в которой предлагаются похожие вещи, которые могут заинтересовать клиента (рис.2). Над этой системой рекомендаций и будет проводиться работа.

Рис. 1. Страница товара и его описание

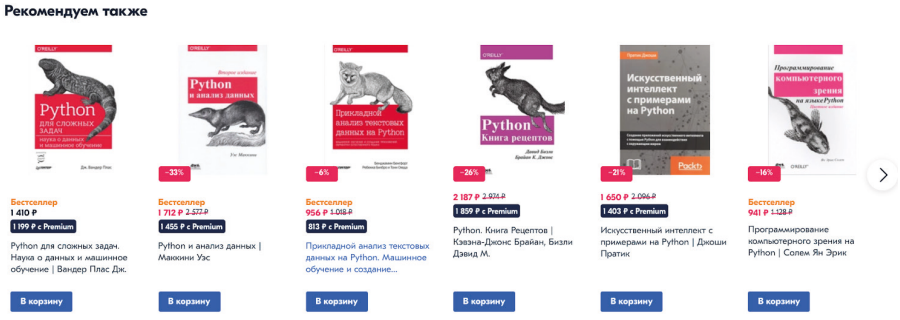

Рис. 2. Рекомендации к просмотренному товару

Система рекомендаций должна показывать, какие товары будут для покупателя наиболее актуальны. Насколько товар подходит пользователю, определяется данными о продажах товара и по истории покупок пользователя. Зная эту информацию, можно реализовать модель предсказания покупок, которые будут рекомендоваться клиенту. Полученный прогноз будет использоваться в качестве критерия для ранжирования, чтобы наиболее релевантные товары показывались первыми.

Определение рекомендаций товаров сводится к решению задачи ранжирования. Главным объектом является тройка элементов, на основании которого будет рассчитываться вероятность покупки:

$$X = \{c, i, r\}_{k=1}^{N},$$

где c – клиент, i – товар, который смотрит клиент, r – рекомендуемый товар.

Далее для каждого клиента (c) и товара (i) формируется свой список рекомендаций:

$$(c, i) \to r.$$

Для улучшения продаж рекомендуемых товаров необходимо, чтобы на первых местах в списке рекомендаций стояли те товары, которые имеют наибольшую вероятность покупки для данной пары клиента и товара, который смотрит клиент. Поэтому необходимо отсортировать данный список по уменьшению вероятности покупки товара:

$$r = \{r_1 \succeq r_2 \succeq r_3 \succeq \ldots \succeq r_n\},$$

где $1, \ldots, n$ – количество рекомендуемых товаров в списке.

На сайте уже реализована своя модель рекомендаций, она достаточно хорошо работает. Полное изменение данной системы не является целью данной работы. Основная задача – улучшение прогноза модели путем преобразования исходных данных и разработки дополнительных признаков, а также выбора более подходящего алгоритма модели. Прогноз строится для каждой пары клиента и просмотренного товара и предсказывается вероятность покупки каждого из товаров списка рекомендаций. В дальнейшем это поможет получить более точные рекомендации.

3. РЕШЕНИЕ ПОСТАВЛЕННОЙ ЗАДАЧИ

В качестве базовой модели использовалась модель бинарной классификации Случайный лес (Random Forest). Для обучения и тестирования использовались исторические данные клиентов. Учитывалось, какие товары смотрел покупатель, что добавлял к себе в корзину, что рекомендовалось, был ли куплен рекомендованный товар или нет, а также рассчитывались дополнительные признаки в виде описательных метрик для рекомендованного товара. Ниже представлены некоторые из них:

- количество просмотров/добавлений в корзину рекомендованного товара;
- количество просмотров/добавлений в корзину рекомендованного товара в первый и последний день наблюдаемого периода;
- конверсия добавления в корзину рекомендованного товара (отношение количества добавлений в корзину к количеству просмотров).

Используя данную модель, было получено базовое качество прогнозов. Ориентируясь на него, необходимо было преобразовать и дополнить исходный набор данных, чтобы улучшить качество предсказаний.

В ходе работы исходные данные были дополнены историей пользователей в течении одного посещения интернет-магазина и текстовыми описаниями товаров. Для этих данных были разработаны новые признаки и реализованы несколько стандартных подходов к построению рекомендаций. Их можно поделить на следующие типы:

1) коллаборативные методы фильтрации;
2) вычисление похожести покупателей по их сессиям;
3) вычисление схожести товаров по текстовому описанию;
4) выявление популярности и новизны товаров.

Вычисляя все перечисленные выше признаки, основной объект (клиент, просмотренный товар, рекомендуемый товар) рассматривается комплексно, с разных сторон. Это добавляет в данные дополнительную информацию, отражающую различные зависимости в объектах. Это может помочь модели легче выявлять закономерности покупки относительно текущего товара и рекомендуемого.

После построения всех необходимых признаков проводилось обучение модели и сравнение полученного качества с предсказаниями старого решения.

4. РЕАЛИЗАЦИЯ МЕТОДОВ КОЛЛАБОРАТИВНОЙ ФИЛЬТРАЦИИ

Одним из основных базовых подходов к реализации рекомендаций является метод коллаборативной фильтрации [5]. В нем можно выделить два метода:

1) основанный на сходстве пользователей (user-based);
2) основанный на сходстве элементов (item-based).

Оба метода позволяют находить похожих пользователей по их взаимодействию с объектами и на основании их оценок подбирать подходящие рекомендации для рассматриваемого пользователя, но выявляют эту похожесть немного по-разному. Так,

метод сходства пользователей основывается на схожести самих клиентов. Этот подход определяет, купит ли пользователь рекомендованный товар, если пользователи, похожие на него, покупали этот товар. Метод сходства элементов рассчитывает похожесть самих товаров. Пользователь купит рекомендованный товар, исходя из того, насколько этот товар похож на те товары, которые покупались этим клиентом раньше.

Алгоритм построения user-based подхода основывается на работе с несколькими матрицами. Первая и основная матрица – это матрица взаимодействий клиента и товара, где каждая строка – это идентификатор покупателя, а столбец – идентификатор товара. В ней отражается, какие товары покупал каждый пользователь (рис.3).

R =

clientid \ itemid	0	1	2	3	4
0	1	0	0	1	0
1	0	1	0	1	1
2	0	1	0	1	0
3	1	1	0	0	1

Рис. 3. Матрица взаимодействий клиентов и товаров

Чтобы подобрать товары, которые покупали похожие пользователи, необходимо найти схожесть всех покупателей. Для этого использовалась стандартная мера косинусной схожести [6] между двумя объектами-векторами:

$$\text{similarity} = \cos(\theta) = \frac{\vec{\tilde{n}_0} \cdot \vec{\tilde{n}_1}}{\left\|\vec{\tilde{n}_0}\right\| \left\|\vec{\tilde{n}_1}\right\|},$$

где $\vec{\tilde{n}_0} \cdot \vec{\tilde{n}_1}$ – скалярное произведение векторов клиентов с id, равными 0 и 1 соответственно, $\left\|\vec{\tilde{n}_0}\right\| \left\|\vec{\tilde{n}_1}\right\|$ – произведение евклидовых норм векторов $\vec{\tilde{n}_0}$ и $\vec{\tilde{n}_1}$. Эти векторы представлены в виде строк матрицы взаимодействий на рис.3.

При расчете похожести всех клиентов мы получаем матрицу вида (рис.4):

C =

clientid	0	1	2	3
0	1	0.4	0.5	0.4
1	0.4	1	0.8	0.33
2	0.5	0.8	1	0.4
3	0.2	0.33	0.4	1

Рис. 4. Матрица похожести клиентов

Чтобы выявить, какие товары могут понравиться рассматриваемому пользователю (например, клиенту с id = 0), необходимо взять вектор строки с id нужного

пользователя из матрицы похожестей клиентов (рис.4) и скалярно перемножить на векторы столбцов товаров из матрицы взаимодействий (рис.3). Скалярным произведением двух векторов называется число, равное произведению модулей векторов, умноженное на косинус угла между векторами [7]:

$$\vec{a} \cdot \vec{b} = \left|\vec{a}\right| \cdot \left|\vec{b}\right| \cos \widehat{\vec{a}\vec{b}}.$$

Так получаются оценки того, что товары понравятся пользователю на основании того, покупали ли эти товары похожие пользователи.

Основанный на сходстве элементов (item-based) подход реализуется аналогичным образом с одним лишь изменением, что вместо матрицы похожести клиентов рассчитывается матрица схожести товаров. Все остальные действия остаются без изменений.

5. НАХОЖДЕНИЕ ПОХОЖЕСТИ ПОКУПАТЕЛЕЙ ПО ИХ ПОВЕДЕНИЮ НА САЙТЕ

Еще одной характеристикой пользователя является его поведение во время посещения интернет-магазина. Пусть такое посещение называется сессией. В ней отражается, какие товары клиент смотрел или добавлял в корзину с максимальным перерывом между действиями в 30 минут. Логика расчета схожести товаров в зависимости от пользовательской сессии заключается в том, что если одни и те же товары часто будут смотреться в течение одного посещения, то скорее всего эти товары похожи и их можно вместе рекомендовать. Для нахождения похожести объектов также будем использовать меру косинусной схожести. Для этого составляется матрица взаимодействия товаров в сессиях (рис.5), где отражается, какие товары попадали в какие сессии.

itemid \ sessionid	0	1	2	3
0	1	0	1	0
1	1	0	0	0
2	0	1	1	0
3	1	1	1	1
4	0	0	1	0

Рис. 5. Матрица взаимодействия товаров и пользовательских сессий

Далее попарно для всех строк этой матрицы (рис.5) применяется формула меры косинусной схожести, получая при этом матрицу похожести всех товаров в разрезе пользовательских сессий.

6. РАСЧЕТ СХОЖЕСТИ ПО ТЕКСТОВОМУ ПРЕДСТАВЛЕНИЮ

В полученных данных было представлено текстовое описание товаров: название товара и его описание. Поэтому было принято решение рассчитать косинусную схожесть по текстовому описанию.

Для того чтобы рассчитать косинусную схожесть, необходимо провести предобработку данных. Для этого ко всем имеющимся описаниям товаров применим основы NLP (natural language processing) при работе с текстом:

- токенизация – разбиение текста на токены, в данном случае на слова;
- обработка текста с помощью регулярных выражений;
- лемматизация – приведение слов к их нормальной форме;
- удаление стоп-слов;
- расчет TF-IDF.

TF-IDF – статистическая мера, которая показывает важность слова для конкретного документа [14].

$$\mathrm{tf}\left(t,d\right)-\mathrm{idf}\left(t,d,D\right)=\mathrm{tf}\left(t,d\right)\cdot\log\left(\frac{D}{df_t}\right)$$

tf (t,d) – частота слова t в документе d;

df_t – количество документов, содержащих слово t;

D – общее количество документов;

idf (t,d,D) – инверсия частоты встречаемости слова t в документах D.

Обработанные документы были преобразованы в векторы с помощью векторизованной модели из библиотеки sklearn. Полученные векторы представляли собой разреженную матрицу, содержащую веса для каждого слова каждого документа, имеющего размер $D \cdot n$, где D – количество документов, а n – количество признаков (уникальных слов). Теперь эти веса из матрицы использовались в качестве признака для каждого документа, а сходство между документами вычислялось с использованием косинусного сходства.

7. ПОСТРОЕНИЕ МОДЕЛИ

После подготовки набора данных для обучения и тестирования необходимо начинать использовать модель, настраивать параметры и улучшать качество.

Для решения задачи ранжирования необходимо сначала решить задачу бинарной классификации, в которой необходимо будет предсказывать вероятность добавления рекомендованного товара в корзину. Получив вектор вероятностей, появится возможность отсортировать товары в рекомендациях по убыванию.

Обучение происходило с помощью библиотеки CatBoost. Это метод машинного обучения, основанный на градиентном бустинге (англ. *gradient boosting*). Бустинг – это подход построения композиций, в рамках которого: базовые алгоритмы строятся

последовательно, каждый последующий алгоритм строится таким образом, чтобы исправить ошибки уже построенной композиции. Композиция – объединение N алгоритмов $b_1(x),\ldots,b_n(x)$ в один. Идея композиции заключается в том, чтобы сначала обучить N базовых алгоритмов, а затем усреднить полученные ответы. Градиентный бустинг является одним из лучших способов направленного построения композиции [8]. Главным преимуществом данной библиотеки является то, что она одинаково хорошо работает как с числовыми признаками, так и с категориальными. Данная библиотека имеет хорошую документацию, обширный функционал и проста в использовании, так как не требует особой подготовки модели [9].

Подбор гиперпараметров модели осуществлялся с помощью метода GridSearchCV из библиотеки sklearn [10].

8. МЕТРИКИ

Mean average precision ($map@K$) – одна из наиболее часто используемых метрик качества ранжирования. В $p@K$ и $ap@K$ качество ранжирования оценивается для отдельно взятого объекта. Идея $map@K$ заключается в том, чтобы посчитать $ap@K$ для каждого объекта и усреднить [11]:

$$map@K = \frac{1}{N}\sum_{j=1}^{N} ap@K_j,$$

где N – количество объектов.

Precision at K ($p@K$) – точность на K элементах:

$$p@K = \frac{\sum_{k=1}^{K} r^{\text{true}}(\pi^{-1}(k))}{K} = \frac{\text{количество релевантых элементов}}{K}.$$

Под $\pi^{-1}(k)$ понимается элемент, который в результате перестановки π оказался на k-ой позиции, r^{true} – принимает значения 0 и 1, в зависимости от релевантности элемента.

Average precision at K ($ap@K$) – среднее только для релевантных товаров:

$$ap@K = \frac{1}{K}\sum_{k=1}^{K} r^{\text{true}}\left(\pi^{-1}(k)\right)\cdot p@k,$$

где k – количество только релевантных элементов.

Recall at K ($recall@k$) – доля релевантных элементов, найденных в топ-k рекомендациях [12]:

$$recall@k = \frac{\text{рекомендованные } k \text{ товаров, которые релевантны}}{\text{общее количество релевантных товаров}}.$$

AUC ROC (площадь под кривой ошибок) – доля пар объектов вида (объект класса 1, объект класса 0), который алгоритм верно упорядочил, т.е. первый объект идет в упорядоченном списке раньше [15].

9. АНАЛИЗ РЕЗУЛЬТАТОВ

Важным моментом в анализе обученной модели, является оценка важности признаков. Важность признаков нужна для понимания своего алгоритма, почему он именно так определяет ответ. С помощью важности признаков можно узнать, какие признаки лучше всего влияют на результат модели [13]:

$$\text{feature importance} = \sum_{\text{trees, leafs}_F} (v_1 - \text{avr})^2 \cdot c_1 + (v_2 - \text{avr})^2,$$

$$\text{avr} = \frac{v_1 \cdot c_1 + v_2 \cdot c_2}{c_1 + c_2},$$

где c_1, c_2 представляют собой общий вес объектов в левом и правом листьях. Вес равен количеству объектов в каждом листе, если веса не были заданы; v_1, v_2 представляют собой значения формулы в левом и правом листьях.

В ходе работы над данными, было разработано 22 признака. На следующем рисунке (рис.6) представлена важность этих признаков для построенной модели. На оси абсцисс расположены значения важности признаков, а на оси ординат представлены названия признаков. Можно заметить, что некоторые признаки несут больше информации для модели, чем другие. Например, признак same_items_on_session_view является лучшим признаком, а us_based_view не несет никакой значимости для модели. Эти признаки являются значением схожести рекомендованного товара с просмотренным в зависимости от поведения клиента за сессию и схожести, по основанной на сходстве пользователей (user-based), соответственно.

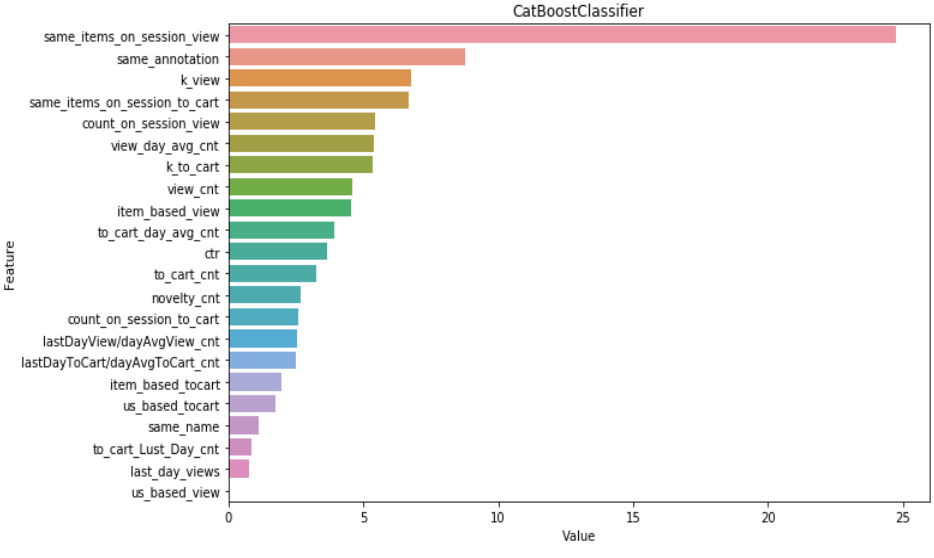

Рис. 6. Важность признаков

В качестве базовой модели был выбран алгоритм RandomForest, обученный на небольшом количестве признаков. На рисунке (рис.7) можно заметить, что обученная модель CatBoostClassifier сильно превзошла базовую модель.

Метрики	Baseline RandomForest	CatboostClassifier
AUC	0.55339	0.6345
Map@3	0.11874	0.1368
Recall@3	0.47102	0.5372

Рис. 7. Анализ результатов. Сравнение с базовой моделью на отложенной выборке

10. ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены возможные методы улучшения прогнозирования вероятности факта покупки рекомендованных товаров на примере интернет-магазина Ozon. В качестве исходных данных использовались данные об истории покупок клиентов, их поведения за сессию и текстовые описания товаров. Для построения прогнозов покупки рекомендованных товаров были использованы модели Random Forest и CatBoostClassifier. Для улучшения прогнозов были разработаны дополнительные признаки для обучения, а для сравнительного анализа моделей были реализованы специальные метрики, которые часто используются для оценки качества рекомендаций. В рамках задачи наилучшие результаты показал градиентный бустинг в реализации CatBoostClassifier.

Литература

1. *Francesco Ricci and Lior Rokach and Bracha Shapira*. Introduction to Recommender Systems Handbook // Springer Science+Business Media, LLC 2011. С. 1–10.
2. *Mizzaro Stefano*. Relevance: The Whole History // journal of the american society for information science, 1997. С. 810–820.
3. *Brent Smith and Greg Linden*. Two Decades of Recommender Systems at Amazon.com // the IEEE Computer Society, 2017. С. 10–17.
4. *Carlos A. Gomez-Uribe and Neil Hunt*. The Netflix Recommender System: Algorithms, Business Value, and Innovation // ACM Transactions on Management Information Systems, Vol. 6, No. 4, Article 13, 2015. С. 6–7.
5. *Е.Е. Пятикоп*. Исследование метода коллаборативной фильтрации на основе сходства элементов // Наукові праці ДонНТУ випуск 2 (18), Серія “Інформатика, кібернетика та обчислювальна техніка”, 2013. С. 109–110.
6. *Кристофер Д. Маннинг, Прабхакар Рагхаван, Хайнрих Шютце*. Введение в информационный поиск // Издательский дом “Вильямс”, 2011. С.138.
7. *Г.Г. Литова, Д.Ю. Ханукаева*. Основы векторной алгебры // Москва, 2009. С. 57.
8. *Jerome H. Friedman.* Greedy Function Approximation: A Gradient Boosting Machine // Technical Discussion: Foundations of TreeNet(tm), 1999. С. 39.
9. *CatBoost* [Электронный ресурс] // URL: https://neerc.ifmo.ru/wiki/index.php?title=CatBoost

10. *GridSearchCV* [Электронный ресурс] // Scikit-learn URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
11. *Gunnar Schröder*, *Maik Thiele, Wolfgang Lehner*. Setting Goals and Choosing Metrics for Recommender System Evaluations, 2011 С. 8.
12. *Ziwei Zhu, Jianling Wang, James Caverlee* // Improving Top-K Recommendation via Joint Collaborative Autoencoders, IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4 License, 2019. С. 7.
13. CatBoost Feature Importance [Электронный ресурс] // catboost URL: https://catboost.ai/docs/concepts/fstr.html#fstr
14. *Wen Zhang, Taketoshi Yoshida, Xijin Tang*. A comparative study of TFIDF, LSI and multi-words for text classification // Expert Systems with Applications, 2010. С. 8.
15. *Tom Fawcett*. An introduction to ROC analysis // Pattern Recognition Letters 27, 2006. С. 865.

Prediction the Probability of Purchases Recommended Items

Pavel A. Parfenov*
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0001-5995-347X
e-mail: pentalbymf@mail.ru

Alena A. Timofeeva**
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0001-7043-3715
e-mail: alena195101@yandex.ru

Gleb B. Sologub***
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0002-5657-4826
e-mail: glebsologub@ya.ru

Andrey S. Alekseychuk****
Moscow Aviation Institute (National Research University), Moscow, Russia

This paper discusses various methods for improving recommendation systems. A comparative analysis of two models for solving classification problems is performed: random forest and CatBoostClassifier. The research was performed on the data of the purchase history of Ozon customers. Standard methods that are often used in recommendation systems were used. We implemented collaborative filtering methods, cosine similarity of products from customer views per site visit, and similarity of text data. To evaluate the results, we used special metrics that evaluate the quality of predictions of the first k objects from the recommendations: Mean average precision (map@K) and Recall at K (recall@k). When generating additional features based on various methods that reveal the similarity of objects, an increase in the quality of model forecasts is noted. The CatBoostClassifier model showed the best results.

Keywords: recommendation systems, machine learning, binary classification, collaborative filtering methods, cosine similarity, map@K, recall@k.

For citation:
Parfenov P.A., Timofeeva A.A., Sologub G.B., Alekseychuk A.S.. Prediction the Probability of Purchases Recommended Items. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2020. Vol. 10, no. 4, pp. 17–30. DOI: https://doi.org/10.17759/mda.2020100402 (In Russ., abstr. in Engl.).

***Pavel A. Parfenov,** Moscow Aviation Institute (National Research University), Moscow, Russian Federation, ORCID: https://orcid.org/0000-0001-5995-347X, e-mail: pentalbymf@mail.ru

****Alena A. Timofeeva.,** Moscow Aviation Institute (National Research University), Moscow, Russian Federation, ORCID: https://orcid.org/0000-0001-7043-3715, e-mail: alena195101@yandex.ru

*****Gleb B. Sologub,** Moscow Aviation Institute (National Research University), Moscow, Russian Federation, ORCID: https://orcid.org/0000-0002-5657-4826, e-mail: glebsologub@ya.ru

******Andrey S. Alekseychuk,** Moscow Aviation Institute (National Research University), Moscow, Russian Federation

References

1. Francesco Ricci and Lior Rokach and Bracha Shapira. Introduction to Recommender Systems Handbook// Springer Science+Business Media, LLC 2011, pp. 1–10.
2. Mizzaro Stefano. Relevance: The Whole History // journal of the american society for information science, 1997, pp. 810–820.
3. Brent Smith and Greg Linden. Two Decades of Recommender Systems at Amazon.com // the IEEE Computer Society, 2017, pp. 10–17.
4. Carlos A. Gomez-Uribe and Neil Hunt. The Netflix Recommender System: Algorithms, Business Value, and Innovation // ACM Transactions on Management Information Systems, Vol. 6, No. 4, Article 13, 2015, pp. 6–7.
5. E.E. Pyatikop. Study of the method of collaborative filtering based on the similarity of elements // Naukovi Pratsi DonNTU vipusk 2 (18), Series “Informatika, Kibernetika TA obchislyuvalna Tehnika”, 2013, pp. 109–110.
6. Christopher D. Manning, Prabhakar Raghavan, Heinrich schütze. Introduction to information retrieval // Publishing house “Williams”, 2011, pp. 138.
7. G.G. Litova, D.Y. Khanukaeva, Basics of vector algebra, Moscow, 2009, pp. 57.
8. Jerome H. Friedman. Greedy Function Approximation: A Gradient Boosting Machine // Technical Discussion: Foundations of TreeNet(tm), 1999. P. 39.
9. CatBoost [Electronic resource] // URL: https://neerc.ifmo.ru/wiki/index.php?title=CatBoost
10. GridSearchCV [Electronic resource] // Scikit-learn URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
11. Gunnar Schröder, Maik Thiele, Wolfgang Lehner. Setting Goals and Choosing Metrics for Recommender System Evaluations, 2011 P. 8.
12. Ziwei Zhu, Jianling Wang, James Caverlee // Improving Top-K Recommendation via Joint Collaborative Autoencoders, IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4 License, 2019. P. 7.
13. CatBoost Feature Importance [Electronic resource] // catboost URL: https://catboost.ai/docs/concepts/fstr.html#fstr
14. Wen Zhang, Taketoshi Yoshida, Xijin Tang. A comparative study of TFIDF, LSI and multi-words for text classification // Expert Systems with Applications, 2010. P. 8.
15. Tom Fawcett. An introduction to ROC analysis // Pattern Recognition Letters 27, 2006. P. 865.

Моделирование и анализ данных
2020. Том 10. № 4. С. 31–40
DOI: https://doi.org/10.17759/mda.2020100403
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2020. Vol 10, no. 4, pp. 31–40
DOI: https://doi.org/10.17759/mda.2020100403
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

УДК 004.85

Использование методов машинного обучения для решения задач прогнозирования суммы и вероятности покупки на основе данных электронной коммерции

Мамиев О.А.*
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0003-1137-4019
e-mail: olegios@mail.ru

Финогенов Н.А.**
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0001-7680-9496
e-mail: finogenov.nik@gmail.com

Сологуб Г.Б.**
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0002-5657-4826
e-mail: glebsologub@ya.ru

Для цитаты:
Мамиев О.А., Финогенов Н.А., Сологуб Г.Б. Использование методов машинного обучения для решения задач прогнозирования суммы и вероятности покупки на основе данных электронной коммерции // Моделирование и анализ данных. 2020. Том 10. № 4. С. 31–40. DOI: https://doi.org/10.17759/mda.2020100403

***Мамиев Олег Аланович,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0003-1137-4019, e-mail: olegios@mail.ru

****Финогенов Никита Андреевич,** Московский авиационный институт (национальный исследовательский университет), г.Москва, Российская Федерация, ORCID: https://orcid.org/0000-0001-7680-9496, e-mail: finogenov.nik@gmail.com

*****Сологуб Глеб Борисович,** кандидат физико-математических наук, доцент кафедры, Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0002-5657-4826, e-mail: glebsologub@ya.ru

Работа направлена на исследование возможности применения методов машинного обучения для построения моделей прогнозирования вероятности покупки и суммы покупки клиентов интернет магазинов. Предлагаются к рассмотрению ранее не используемые в рамках конкретной задачи подходы к решению. В качестве выборки использованы данные о транзакциях пользователей сайта ponpare.jp в период с 01.07.2011 по 23.06.2012. Приводится описание и сравнительный анализ наиболее распространенных методов решения аналогичных задач. Описываются метрики, использованные для оценки результатов в случае прогнозирования факта и суммы покупки. Полученные результаты дают понять, что в рамках задачи предсказания вероятности покупки градиентный бустинг, а именно его реализация LGBMClassifier, показывает наиболее точную оценку. Для задачи прогнозирования суммы покупки клиента использование градиентного бустинга также дало наилучшие результаты.

Ключевые слова: прогноз вероятности и суммы покупки, классификация, регрессия, анализ данных, обработка данных, машинное обучение

1. ВВЕДЕНИЕ

В наше время проблема «Больших данных» является основной как с точки зрения научных исследований, так и с точки зрения бизнес-аналитики, для понимания и оптимизации бизнес процессов. Данные широко используются для решения различных задач, возникающих в повседневной жизни – это и рекомендация товаров в интернет – магазинах, и выдача результатов в поисковых движках, и обработка данных с камер наблюдения, и создание беспилотных автомобилей и т.д.

Данная работа посвящена решению задач прогнозирования вероятности и суммы покупки клиентов сайта ponpare.jp. Решение данных проблем является крайне актуальной задачей.

Во-первых, цифровые платформы нацелены на оптимизацию видимости и целевого маркетинга своих продуктов. Такая стратегия не является чем-то новым и уже много лет реализуется в физических магазинах. Исследования поведения клиентов и попытки предсказать их намерения предпринимались еще в конце прошлого века [1]. Улучшенное размещение продуктов в супермаркетах приводит к увеличению их видимости, что в свою очередь обуславливает увеличение продаж. Аналогичные концепции могут быть применены и для интернет-магазинов. Умение заранее распознать намерение клиента приобрести товар (и как следствие улучшенный целенаправленный персональный маркетинг) может снизить затраты и повысить эффективность работы компании.

Во-вторых, почти вся часть современной экономики основана на информации и по этому различные виды электронной коммерции (ecommerce) становятся самыми востребованными и уверенно вытесняют физическую коммерцию, а торговля непосредственно между клиентами (в виде торговых интернет площадок) (B2B ecommerce), становиться наиболее распространенной. Так в 2015 оборот B2B ecommerce составил 21.5 миллиона евро и Россия занимала первое место по количеству электронных покупателей, и их количество неуклонно растет [2]. Подобное развитие интернет-

торговли в том числе обусловлено желанием компаний понимать потребности покупателей и их финансовые возможности. Знание о средней суммы покупки пользователя позволяет оптимизировать цену предоставляемой услуги, скидки, акции и т.д.

Существуют различные методы машинного обучения, применяемые для аналогичных задач. В данной статье проводится сравнительный анализ основных и представляется подход к решению задач прогнозирования вероятности и суммы покупки.

2. ОПИСАНИЕ ДАТАСЕТА

В качестве исходного датасета были использованы данные о транзакциях 22873 клиентов сайта ponpare.jp в период с 01.07.2011 по 23.06.2012. Они структурированы в шести таблицах.

1. Данные о пользователях (22873) содержат следующие атрибуты: идентификатор пользователя, возраст, пол, дата регистрации, дата удаления аккаунта (если происходило), область проживания;
2. Данные о купонах (19413) содержат такие атрибуты как: идентификатор купона, название категории и подкатегории, размер скидки, цена без скидки, цена со скидкой, начало действия купона, окончания действия купона, дата начала и конца скидки, возможность использования купона в конкретный день недели и праздничный день, название области, в которой купон можно использовать;
3. Журнал с просмотрами купонов (2833180) содержит: статус покупки (состоялась она или нет), идентификатор покупки, дата просмотра, идентификатор купона, идентификатор пользователя, идентификатор сессии;
4. Журнал с покупками пользователей (168996) включает информацию о: идентификаторах пользователя, купона и покупки, количестве приобретенных товаров, дате произведения покупки, короткое название области;
5. Данные об областях действия купонов (138185) содержат: идентификатор купона, короткое название области и название префектуры;
6. Данные о местоположении областей (47) содержат: название области, широту, долготу.

3. ПРОГНОЗИРОВАНИЕ ФАКТА ПОКУПКИ КЛИЕНТА

Решение задачи прогнозирования факта покупки сведем к решению задачи бинарной классификации.

На вход получаем набор признаков $(x_1, ..., x_n)$, а на выход выдаем вектор y прогноза факта покупки, то есть бинарное значение, соответствующее одному из двух классов. Первый включает в себя потенциальных покупателей, а второй – пользователей, не планирующих приобретать товар.

Проанализируем наиболее распространенные методы решения подобных задач.

Логистическая регрессия (Logistic Regression) – алгоритм классификации, используемый для определения вероятности успеха и неудачи события. Он поддерживает категоризацию данных по дискретным классам путем изучения взаимосвязи

из заданного набора помеченных данных. Изучается линейная зависимость из заданного набора данных, а затем вводится нелинейность в форме сигмоидной функции.

Достоинства данного алгоритма заключаются в легкости реализации, интерпретации и эффективности при достаточно быстрой скорости обучения. Однако, если количество наблюдений меньше, чем количество признаков, логистическую регрессию использовать не следует, иначе это может привести к переобучению. К тому же, она способна строить только линейные границы.

Деревья решений (Decision trees) – алгоритм, решающий проблему машинного обучения путем преобразования данных в представление дерева. Каждый внутренний узел древовидного представления обозначает атрибут, а каждый листовой узел обозначает метку класса. По сравнению со многими другими алгоритмами деревья решений требуют меньше усилий для подготовки данных во время предварительной обработки. Для обучения нет необходимости нормализовывать или масштабировать данные. К сожалению, даже небольшое изменение данных может вызвать существенное изменение структуры дерева решений, что приводит к нестабильности. Сами вычисления могут быть намного более сложными по сравнению с другими алгоритмами и зачастую занимать много времени.

Алгоритм KNN (K-Nearest Neighbors) предполагает, что похожие вещи существуют в непосредственной близости. Другими словами, похожие вещи находятся рядом друг с другом. KNN достаточно прост, эффективен и интуитивен, но несмотря на это все же имеет несколько ограничений. При достаточно большом тренировочном наборе алгоритм может иметь длительное время выполнения. Он очень чувствителен к несущественным или избыточным признакам, однако при тщательном применении отбора признаков (feature selection) или взвешивания признаков (feature weighting) этого можно избежать. Кроме того, при обучении, основанном на дистанциях, не всегда понятно в чем эту дистанцию измерять с целью добиться наилучших результатов.

Градиентный бустинг (Gradient Boosting) – крайне популярный алгоритм, способный решать задачи классификации, регрессии и ранжирования. В качестве модели градиентного бустинга был использован LightGBM. LightGBM – это фреймворк, в котором используется алгоритм обучения основанный на деревьях решения [3]. Это распределенный и очень эффективный метод для решения задач классификации и регресии . К его достоинствам можно отнести следующие:

1. Быстрая скорость обучения и высокая эффективность, LightGBM использует подход на основе гистограммы (histogram-based) [4], преобразуя непрерывные значения признаков в дискретные.
2. Малый объем памяти.
3. Более высокая точность (в сравнение с другими моделями бустинга): LightGBM создает более сложное дерево, чем метод поэтапного разделения, вследствие чего достигается более высокая точность. Однако это может привести к переобучению, и поэтому необходимо грамотно подходить к глубине дерева.
4. Возможность обработки данных большого размера: в сравнении с другой популярной моделью XGBoost [5], LightGBM позволяет построить более точную модель из-за сокращения времени обучения [6].

4. ПРОГНОЗИРОВАНИЕ СУММЫ ПОКУПКИ КЛИЕНТА

Сформулируем задачу прогнозирования суммы покупки клиента как построение модели машинного обучения, которая позволит построить вектор ответов y (сумма, которую клиент тратит на покупку купонов) в зависимости от набора признаков $(x_1, ..., x_n)$ (данным о клиенте и транзакциях):

$$y = f(x_1, ..., x_n) + \varepsilon,$$

где ε – вектор отклонений модельных данных от исходных.

Цель – используя обучающие данные построить функцию $\hat{f}(x_1, ..., x_n)$, которая могла бы служить аппроксимацией для функции $f(x_1, ..., x_n)$. Существует множество способов для успешного решения поставленной задачи: полиномиальная линейная регрессия (Ordinary Linear Regression) [7], частичная полиномиальная линейная регрессия (Partial Least Squares Regression) [8], метод опорных векторов (support vector regression) [9], нейронные сети [10], модели бустинга [11].

В нашем случае решено было использовать следующие подходы:

1. Полиномиальная линейная регрессия.
2. Градиентный бустинг (Gradient Boosting).

Использование этих методов имеет свои преимущества и недостатки. К достоинствам многомерные регрессионные модели (multivariate regression models) можно отнести простоту реализации и быстроту обучения модели, что позволяет проводить большое количество экспериментов. Вследствие чего использование такой простой, хоть и не очень точной модели, предоставляет возможность проверять различные гипотезы о структуре данных, генерировать новые признаки, производить отбор признаков. В данной работе программной реализации была использована модель LinearRegression из пакета sklearn. Для получения итогового результата был использован LightGBM.

5. МЕТРИКИ ОЦЕНКИ КАЧЕСТВА

Для задачи прогнозирования вероятности покупки клиента в основу были выбраны такие метрики, как: precision, recall и f-score.

Точность (precision) в пределах класса – это доля объектов действительно принадлежащих данному классу относительно всех объектов, которые модель определила в этот класс:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

где TP – истинно – положительное решение, TN – истинно – отрицательное решение, FP -ложно – положительное решение, FN – ложно – отрицательное решение.

Полнота (recall) – это доля найденных объектов, принадлежащих к данному классу, относительно всех объектов из этого класса в тестовой выборке:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

К сожалению, в реальной жизни максимальная точность и полнота одновременно бывают достижимы крайне редко. Поэтому, хотелось бы опираться на метрику, превносившую некий баланс, объединяя в себе информацию о точности и полноте.

F1 -мера (f1-score) определяется как взвешенное гармоническое среднее значение точности и отзыва теста:

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}.$$

Для задачи прогнозирования суммы покупки клиента были выбраны RMSE, R^2_{score}.

RMSE (Среднеквадратичная ошибка) – это мера того, насколько близка оценка к фактическим данным:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2},$$

где $\hat{y}$ – значения полученные моделью, y – реальные значения. Чем меньше RMSE, тем лучше предсказание модели.

R^2_{score} (коэффициент детерминации) – показывает близость предсказанной модели к реальной модели. Для расчета необходимо вычислить TSS (общая сумма квадратов отклонений – число равное сумме квадратов разности элементов выборки и среднего) и SSE (сумма квадратов невязок – число равное сумме квадратов отклонений модельных данных от исходных):

$$\mathrm{TSS} = \sum_{i=1}^{n}(\bar{y} - y_i)^2,$$

$$\mathrm{SSE} = \sum_{i=1}^{n}(\widehat{y_i} - y_i)^2,$$

где $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Тогда R^2_{score}:

$$R^2_{score} = 1 - \frac{\mathrm{SSE}}{\mathrm{TSS}}.$$

R^2_{score} принимает значения от 0 до 1 и чем выше значение, тем точнее полученная модель.

6. АНАЛИЗ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТА

В качестве моделей для решения задачи прогнозирования факта покупки товара были выбраны логистическая регрессия, деревья решения, метод k-ближайших соседей и градиентный бустинг.

Model	Precision	Recall	F1-score
Logistic Regression	0.602	0.607	0.604
Decision trees	0.566	0.554	0.56
KNN	0.604	0.619	0.611
LGBM	0.652	0.648	0.65

Рис. 1. Результаты прогнозирования факта покупки

```
Index(['SEX_ID', 'AGE', 'PREF_NAME', 'CAPSULE_TEXT', 'GENRE_NAME',
       'PRICE_RATE', 'CATALOG_PRICE', 'DISCOUNT_PRICE', 'DISPPERIOD',
       'VALIDPERIOD', 'USABLE_DATE_MON', 'USABLE_DATE_TUE', 'USABLE_DATE_WED',
       'USABLE_DATE_THU', 'USABLE_DATE_FRI', 'USABLE_DATE_SAT',
       'USABLE_DATE_SUN', 'USABLE_DATE_HOLIDAY', 'USABLE_DATE_BEFORE_HOLIDAY',
       'large_area_name', 'ken_name', 'small_area_name', 'IS_ACTIVE_USER',
       'target', 'I_DATE_year', 'I_DATE_month', 'I_DATE_dayofweek',
       'REG_DATE_year', 'REG_DATE_month', 'REG_DATE_dayofweek',
       'DISPFROM_year', 'DISPFROM_month', 'DISPFROM_dayofweek', 'DISPEND_year',
       'DISPEND_month', 'DISPEND_dayofweek', 'VALIDFROM_year',
       'VALIDFROM_month', 'VALIDFROM_dayofweek', 'VALIDEND_year',
       'VALIDEND_month', 'VALIDEND_dayofweek'],
```

Рис. 2. Набор признаков для решения задачи прогнозирования суммы покупки

Результаты работы, представленные на рис. 1, дают понять, что реализация градиентного бустинга LGBM выдает наиболее точный ответ одновременно и относительно precision, и recall. Деревья решений же справляются с данным датасетом хуже всего, при этом занимая внушительный промежуток времени в сравнении с прочими моделями.

Для решения задачи прогнозирования суммы покупки были обучены модели multivariate regression и LightGBM с использованием следующих признаков:

В результате имеем следующие значения метрик оценки качества:

Model	RMSE	R^2_{score}
Multivariate regression	4285.964	0.216
LGBM	3947.887	0.334

Рис. 3. Результаты прогнозирования суммы покупки

Рис. 3 показывает, что использование градиентного бустинга для данной задачи, позволяет получить наиболее точный ответ.

7. ВЫВОД

В данной работе были рассмотрены возможности применения методов машинного обучения для прогнозирования вероятности факта и суммы покупки клиентов интернет магазинов. В качестве исходных данных использовались данные о клиентах и их транзакциях с сайта ponpare.jp. Были построены и обучены следующие модели машинного обучения Logistic Regression, Decision trees, KNN, LightGBM Classification для задачи прогнозирования вероятности покупки и multivariate regression, LightGBM Regression для задачи прогнозирования суммы покупки клиента. В рамках обеих задач наилучшие результаты показал градиентный бустинг в реализации LGBM.

Литература

1. *Day, D., Gan, B., Gendall, P. and Esslemont, D.* Predicting purchase behaviour // Marketing Bulletin. 1991. P.18–30.
2. *Starostin, V.S. and CHERNOVA, V.Y.* E-commerce development in Russia: trends and prospects // The Journal of Internet Banking and Commerce. 2016.

3. *Kuhn M, Johnson K.* Applied predictive modeling // New York: Springer. 2013.
4. *Glasbey, C.A*. An analysis of histogram-based thresholding algorithms // CVGIP: Graphical models and image processing. 1993. P. 532–537.
5. https://github.com/dmlc/xgboost
6. *Yang S, Zhang H.* Comparison of several data mining methods in credit card default prediction // Intelligent Information Management. 2018. P. 115.
7. *Wu, H., Jiao, H., Yu, Y., Li, Z., Peng, Z., Liu, L. and Zeng, Z.* Influence factors and regression model of urban housing prices based on internet open access data // Sustainability. 2018. P. 1676.
8. *Liu, L., Ji, M. and Buchroithner, M.* Combining partial least squares and the gradient-boosting method for soil property retrieval using visible near-infrared shortwave infrared spectra // Remote Sensing. 2017. P. 1299.
9. *Wu, J.Y.* Housing Price prediction Using Support Vector Regression. 2017.
10. *Limsombunchai, V.* House price prediction: hedonic price model vs. artificial neural network // In New Zealand agricultural and resource economics society conference. 2004. P. 25–26.
11. *Li, J.Z.* Monthly Housing Rent Forecast Based on LightGBM (Light Gradient Boosting) Model // International Journal of Intelligent Information and Management Science, 2018.

Using Machine Learning Methods to Solve Problems of Forecasting the Amount and Probability of Purchase Based on E-Commerce Data

Oleg A. Mamiev*
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0003-1137-4019
e-mail: olegios@mail.ru

Nikita A. Finogenov**
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0001-7680-9496
e-mail: finogenov.nik@gmail.com

Gleb B. Sologub****
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0002-5657-4826
e-mail: glebsologub@ya.ru

The study is aimed at investigating the possibility of using machine learning methods to build models for predicting the probability of purchase and the amount of purchase by online store customers. As a sample, we used data of users transactions of the site ponpare.jp in the period from 01.07.2011 to 23.06.2012. The description and comparative analysis of the most common methods for solving similar problems are given. The metrics used to measure the results in the case of forecasting the fact and amount of the purchase are being described. The results obtained make it clear that within the framework of the problem of predicting the probability of a purchase, gradient boosting, namely its implementation of LGBMClassifier, shows the most accurate estimate. For the problem of predicting the amount of a customer's purchase, using gradient boosting also gave the best results.

Keywords: probability and purchase amount forecast, classification, regression, data analysis, data processing, machine learning.

For citation:
Mamiev O.A., Finogenov N.A., Sologub G.B. Using Machine Learning Methods to Solve Problems of Forecasting the Amount and Probability of Purchase Based on E-Commerce Data. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2020. Vol. 10, no. 4, pp. 31–40. DOI: https://doi.org/10.17759/mda.2020100403 (In Russ., abstr. in Engl.).

****Oleg A. Mamiev***, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000-0003-1137-4019, e-mail: olegios@mail.ru

*****Nikita A. Finogenov***, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000-0001-7680-9496, e-mail: finogenov.nik@gmail.com

******Gleb B. Sologub***, PhD (Physics and Mathematics), Associate Professor, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000-0002-5657-4826, e-mail: glebsologub@ya.ru

References

1. Day, D., Gan, B., Gendall, P. and Esslemont, D. Predicting purchase behaviour // Marketing Bulletin. 1991. P.18–30.
2. Starostin, V.S. and CHERNOVA, V.Y. E-commerce development in Russia: trends and prospects // The Journal of Internet Banking and Commerce. 2016.
3. Kuhn M, Johnson K. Applied predictive modeling // New York: Springer. 2013.
4. Glasbey, C.A. An analysis of histogram-based thresholding algorithms // CVGIP: Graphical models and image processing. 1993. P. 532–537.
5. https://github.com/dmlc/xgboost
6. Yang S, Zhang H. Comparison of several data mining methods in credit card default prediction // Intelligent Information Management. 2018. P. 115.
7. Wu, H., Jiao, H., Yu, Y., Li, Z., Peng, Z., Liu, L. and Zeng, Z. Influence factors and regression model of urban housing prices based on internet open access data // Sustainability. 2018. P. 1676.
8. Liu, L., Ji, M. and Buchroithner, M. Combining partial least squares and the gradient-boosting method for soil property retrieval using visible near-infrared shortwave infrared spectra // Remote Sensing. 2017. P. 1299.
9. Wu, J.Y. Housing Price prediction Using Support Vector Regression. 2017.
10. Limsombunchai, V. House price prediction: hedonic price model vs. artificial neural network // In New Zealand agricultural and resource economics society conference. 2004. P. 25–26.
11. Li, J.Z. Monthly Housing Rent Forecast Based on LightGBM (Light Gradient Boosting) Model // International Journal of Intelligent Information and Management Science, 2018.

Моделирование и анализ данных
2020. Том 10. № 4. С. 41–50
DOI: https://doi.org/10.17759/mda.2020100404
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2020. Vol 10, no. 4, pp. 41–50
DOI: https://doi.org/10.17759/mda.2020100404
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

УДК 004.85

Использование методов машинного обучения для решения задач прогнозирования спроса на новый товар в интернет-маркетплейсе

Осин А.А.*
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0002-2664-1370
e-mail: artemosin1@yandex.ru

Фомин А.К.**
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0003-3545-4435
e-mail: artem.fomin@outlook.com

Сологуб Г.Б.****
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0002-5657-4826
e-mail: glebsologub@ya.ru

Виноградов В.И.******
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0003-3773-9653
e-mail: vvinogradov@inbox.ru

Для цитаты:
Осин А.А., Фомин А.К., Сологуб Г.Б, Виноградов В.И. Использование методов машинного обучения для решения задач прогнозирования спроса на новый товар в интернет-маркетплейсе // Моделирование и анализ данных. 2020. Том 10. № 4. С. 41–50. DOI: https://doi.org/10.17759/mda.2020100404

***Осин Артем Александрович,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0002-2664-1370, e-mail: artemosin1@yandex.ru

****Фомин Артем Константинович,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0003-3545-4435, e-mail: artem.fomin@outlook.com

Работа направлена на исследование возможности применения методов машинного обучения для построения моделей прогнозирования спроса на новые товары в интернет-магазине Ozon.ru. Предлагаются к рассмотрению ранее неиспользуемые в рамках конкретной задачи подходы к решению. В качестве выборки использованы данные об истории продаж и хранении товаров на сайте Ozon.ru. Приводится описание, анализ примерного убытка сайта Ozon.ru, используемых данных, процесса построения базовой модели, а также полученных результатов. Описываются метрики, использованные для оценки результатов прогнозирования, а также проводится сравнительный анализ между результатами предсказания построенной модели и результатами эвристически подобранных значений.

Ключевые слова: прогнозирования спроса, новый товар, энкодинг, градиентный бустинг, регрессия, препроцессинг, обработка данных, машинное обучение.

****Сологуб Глеб Борисович,*** кандидат физико-математических наук, доцент кафедры, Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0002-5657-4826, e-mail: glebsologub@ya.ru

*****Виноградов Владимир Иванович,*** кандидат физико-математических наук, доцент кафедры, Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0003-3773-9653, e-mail: vvinogradov@inbox.ru

1. ВВЕДЕНИЕ

Всё больше участников рыночной экономики стремятся предоставлять свои товары и услуги онлайн. Поэтому неудивительно, что крайнюю востребованность приобрели интернет-маркетплейсы, которые выступают посредниками между предпринимателями и конечным потребителем.

Маркетплейсы предоставляют услуги предпринимателям, связанные с распространением их продукции, а также имеют возможность предоставлять потребителям свои услуги и товары. И в том, и в другом случае крайне актуальна задача предсказания спроса, так как и у маркетплейса и у предпринимателей, пользующихся услугами маркетплейсов, появляется возможность эффективно управлять денежными ресурсами – закупать только те товары, которые будут хорошо продаваться. То есть и сама платформа, и поставщик смогут лучше реагировать на изменения рынка и получать большую прибыль.

Цель работы заключается в проектировании системы для прогнозирования спроса на новые товары в интернет-маркетплейсе. В частности, необходимо разработать отдельный модуль, задача которого прогнозировать продажи товаров-новинок в зависимости от описательных характеристик и внешней информации.

Сформулируем этапы решения этой задачи:

1. Вычисления примерного убытка интернет-маркетплейса «Озон» для возможности интерпретации результата работы

2. Разработка базовой модели, которая работает с некоторым небольшим количеством признаков каждого товара и позволит оценить качество результатов работы конечной модели, а также необходима для понимания предоставленных данных.
3. Разработка специализированных моделей, для которых будут различаться природа исходных данных, а именно описание товара, изображения товара и данные, полученные из внешних источников.
4. Построение комбинированной модели на основе специализированных моделей.

2. ОПИСАНИЕ ДАТАСЕТА И ПРЕДОБРАБОТКА

Для построения базовой модели были представлены данные интернет-маркетплейсом «Озон», в частности, история продаж товаров, размером 1886147 уникальных продаж, история товаров на складе размером 13632607 уникальных значений и информация о категориях товаров размером 483280 значений.

	categorylevel1	fake_itemid
0	TV, audio, Hi-Fi & other electronics	0
1	TV, audio, Hi-Fi & other electronics	1
2	TV, audio, Hi-Fi & other electronics	2
3	TV, audio, Hi-Fi & other electronics	3
4	TV, audio, Hi-Fi & other electronics	4

Рис. 1. Информация о категориях товара

Информация о категориях включает в себя категорию и уникальный идентификатор товара.

	date	qty	fake_itemid
0	2015-01-05	1	94489280563
1	2015-03-03	1	94489280563
2	2016-10-10	1	94489280563
3	2016-05-19	1	94489280563
4	2016-11-21	1	94489280563
...	...	...	...
1886142	2020-07-22	1	171798711788
1886143	2020-07-22	1	51539627534
1886144	2020-07-20	1	51539627534
1886145	2020-07-22	1	137438973480
1886146	2020-07-20	1	85899365801

Рис. 2. Вид истории товара на складе

История продаж включала в себя дату продажи, количество и уникальный идентификатор товара.

	moment	price	itemdiscount	virtualdiscount	freeqty	fake_itemid
2077	2019-09-01 00:00:00.0	100.0	5.000000	5.000000	11	111669149762
2108	2019-06-06 00:00:00.0	100.0	13.000000	13.000000	14	111669149762
2110	2019-06-11 00:00:00.0	100.0	13.000000	13.000000	9	111669149762
2112	2019-06-12 00:00:00.0	100.0	13.000000	13.000000	8	111669149762
2122	2018-09-03 00:00:00.0	100.0	10.000000	10.000000	6	111669149762
...	...	...	...	...	...	...
13632591	2020-07-25 00:00:00.0	2200.0	50.062500	50.062500	0	171798712100
13632596	2020-07-25 00:00:00.0	800.0	30.125000	30.125000	0	68719496927
13632598	2020-07-25 00:00:00.0	430.0	9.773438	9.773438	0	85899366074
13632600	2020-07-25 00:00:00.0	4086.0	33.343750	33.343750	0	111669169791
13632604	2020-07-25 00:00:00.0	13455.0	23.078125	23.078125	0	111669169794

Рис. 3. Вид истории продаж товара

Перед обучением модели была произведена первичная обработка данных. В данных были убраны аномалии и пропущенные значения. Затем было необходимо обработать категориальные признаки. В контексте решаемой задачи было необходимо преобразовать категорию товара из текстового формата в целочисленный. Для обработки категориальных признаков существуют два основных подхода – one hot encoding [1] и label encoding [2].

Оба алгоритма работают схожим образом – среди обрабатываемых значений находятся уникальные и из этих значений создаются классы для будущего энкодера. Затем каждому значению выдается метка, означающая принадлежность к тому или иному классу. Различие этих двух методов заключается в способе разметки данных на классы. В случае с label encoding метка класса представляет собой целое число, принадлежащее интервалу $(1,...,n-1)$ где n – число уникальных значений признака. То есть результатом преобразования является столбец, содержащий в себе целые числа. One hot encoder, в свою очередь, создает $(1,...,n-1)$ столбцов, содержащих в себе либо нуль, либо единицу. Каждый столбец представляет собой выделенный уникальный класс. Единица в столбце означает принадлежность к классу, нуль – не принадлежность, соответственно.

Поскольку применение подхода one hot encoding приводит к увеличению размерности матрицы признаков и делает матрицу более разреженной, что может негативно сказаться на результатах обучения модели, нами было принято решение использовать label encoding. Предобработанные данные были соединены в итоговый датасет.

Из обработанных ранее данных были выделены следующие признаки:

- Количество конкурирующих товаров внутри группы категориального дерева;
- Цена товара;
- Категория, к которой относится товар;
- Средняя цена в категориальной группе, содержащей товар, на момент появления товара;
- Средние продажи в категориальной группе в прошлом месяце;
- Средние продажи в категориальной группе три месяца назад;
- Средние продажи в категориальной группе шесть месяцев назад;
- Средние продажи в категориальной группе год назад.

После извлечения описанных признаков, в датасете были оставлены только данные о первом появления товара на торговой площадке. В качестве целевого признака для обучаемой модели было взято количество продаж товара за текущий месяц. Полученный датасет был разбит на обучающую и тестовую выборку в соотношении 80/20.

3. ОПИСАНИЕ МОДЕЛИ

В качестве модели была использована LightGBM [3]. Это быстрый, распределенный, высокопроизводительный градиентный бустинг, основанный на деревьях решений. Он часто используется для задач, классификации, регрессии, ранжирования и других задач машинного обучения. Он обладает несколькими преимуществами перед другими градиентными бустингами [6]:

1. Быстрая скорость обучения и высокую эффективность за счет использования подхода роста деревьев в глубину
2. Низкое потребление памяти.
3. Более высокая точность, чем у других алгоритмов градиентного бустинга за счет построения более сложной структуры решающих деревьев. Однако, иногда это может привести к переобучению модели.
4. Совместимость с большими наборами данных. Способен также хорошо работать с большими наборами данных со значительным сокращением времени обучения по сравнению с XGBOOST.

Для оценки качества предсказаний модели, было принято решение использовать следующие метрики:

Средний модуль отклонения (MAE – Mean Absolute Error или MAD – Mean Absolute Deviation):

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m} | a_i - y_i |,$$

где a_i - фактическое, а y_i - предсказанное значение (здесь и далее).

Средний квадрат отклонения (MSE – Mean Squared Error)[5]:

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m} | a_i - y_i |^2 .$$

Коэффициент детерминации (R^2)[4]:

$$R^2 = 1 - \frac{\sum_{i=1}^{m} | a_i - y_i |^2}{\sum_{i=1}^{m} | \overline{y} - y_i |^2}, \qquad \overline{y} = \frac{1}{m} \sum_{i=1}^{m} y_i,$$

где $\overline{y}$ - среднее значение.

Получены следующие результаты.

Табл. 1

Значения метрик обученной модели

R^2	**0.036743**
MSE	**330.08631**
MAE	**2.6352**

Помимо посчитанных метрик было проведено сравнение результатов предсказания с результатами эвристически подобранных значений. В качестве подобранных значений были взяты результаты работы абстрактной модели, предсказывающей для любых входных сигналов одно и то же целое число. Данные модели далее будем называть константными и для простоты записи, далее будем обозначать модель, предсказывающую для любых входных сигналов число i, самим числом i. В табл. 2 показаны результаты оценки точности подобранных значений предсказаний.

Табл. 2

Сравнение значений метрик обученной и константных моделей

	0	**1**	**2**	**3**	**4**	**Модель**
MAE	3.0122	2.0122	2.2842	2.8951	3.6381	2.6352
MSE	51.751	346.726	343.702	342.677	343.653	330.086
R^2	-0.0264	-0.011	-0.0029	-4.4020	-0.0028	0.03674

По большинству метрик базовая модель превосходит эвристически подобранные предсказания. Значение метрик MAE и MSE для предсказаний нашей модели, в большинстве случаев, меньше, чем значение этих метрик для предсказаний константных моделей. Это означает, что средняя и среднеквадратичная ошибка предсказаний обученной модели меньше, чем у константных моделей, следовательно, созданная модель предсказывает точнее и работает лучше. Особенно это видно по метрике R^2, так как она меньше нуля для всех моделей, основанных на эвристически подобранные предсказаниях, а, значит, простое среднее будет давать результат лучше [4].

Табл.3

Анализ процентной ошибки обученной и константных моделей

	0	**1**	**2**	**3**	**4**	**Модель**
Ошибка в %	100 %	51 %	76 %	102 %	143 %	70 %

Дополнительно, для улучшения интерпретируемости результатов, был проведен сравнительный анализ предсказаний построенной модели и эвристически подобранных значений. Анализ проводился, как определение процентного соотношения между фактическими продажами маркетплейса и предсказаниями описанных выше моделей. Процентное соотношение рассчитывалось, как отношение прибыли маркетплейса в рублях и прибыли, рассчитанной на основе предсказаний моделей. Для удобства количество процентов прибыли, потерянное маркетплейсом из-за неверных предсказаний модели, будем называть процентной ошибкой. Результаты проведенного анализа показаны в табл. 3

Как видно из табл. 3, предложенная модель вполне эффективна и состоятельна. То, что константная модель, предсказывающая единицу для любых входных сигналов, имеет меньшую процентную ошибку, чем предложенная модель, объясняется спецификой полученного датасета – в исходном подмножестве товаров средние продажи были близки к единице.

Также была выделена важность признаков:

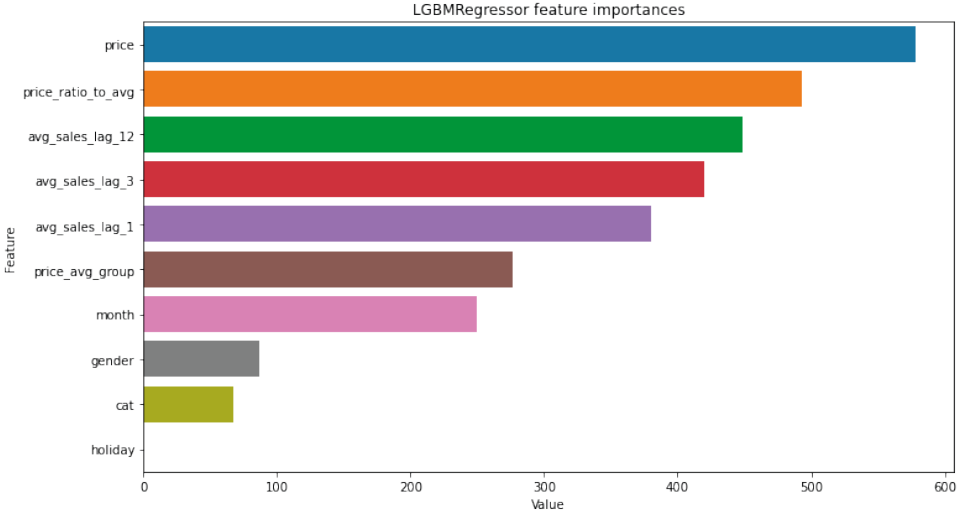

Рис. 4. Значения метрик обученной модели

На рис. 4 представлена диаграмма важности признаков для обученной модели. Данная диаграмма показывает, какими признаками руководствуется алгоритм при создании предсказания в большей или меньшей степени, соответственно. Иными словами диаграмма показывает то, насколько сильно выделенные нами признаки коррелируют с искомым значением. В рассматриваемом случае, анализируя диаграмму, можно убедиться, что наиболее значимыми признаками являются:

- price – цена товара;
- price_ratio_to_avg – отношение цены товара, к средней цене товара в группе;
- avg_sales_lag_12 – средние продажи в группе, к которой принадлежит товар, год назад;

- avg_sales_lag_3 – средние продажи в группе, к которой принадлежит товар, 3 месяца назад;
- avg_sales_lag_1 – средние продажи в группе, к которой принадлежит товар, месяц назад;
- price_avg_group – средняя цена в группе, к которой принадлежит товар;
- month – месяц, в котором товар вышел на торговую площадку;
- cat – категория, к которой принадлежит товар.

4. ВЫВОД

На данном этапе развития работы было представлено два результата: вычислен примерный убыток интернет-маркетплейса «Озон», а также построена базовая модель, обученная на данных, представленными «Озоном». Данная модель будет основой для дальнейших исследований задачи предсказания спроса на товары, не имеющие истории продаж, и будет использоваться в ансамбле вместе с другими моделями, реализующими другие подходы к решению данной задачи.

Литература

1. *Bisong E.* Introduction to Scikit-learn // Building Machine Learning and Deep Learning Models on Google Cloud Platform 2019. P. 215–229.
2. *Cerda P., Varoquaux G., Kégl B.* Similarity encoding for learning with dirty categorical variables // Machine Learning. 2018. P. 1477–1494.
3. *Ke G.* et al. Lightgbm: A highly efficient gradient boosting decision tree // Advances in neural information processing systems. 2017. P. 3146–3154.
4. *Redell N.* Shapley Decomposition of R-Squared in Machine Learning Models // arXiv preprint arXiv:1908.09718. 2019.
5. *Botchkarev, Alexei.* “Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology.” // arXiv preprint arXiv:1809.03006. 2018.
6. *Al Daoud E.* Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset // International Journal of Computer and Information Engineering. 2019. P. 6–10.

Using Machine Learning Methods to Solve Problems of Forecasting Demand for New Products in the Internet Marketplace

Artem A. Osin*
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0002-2664-1370
e-mail: artemosin1@yandex.ru

Artem K. Fomin**
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0003-3545-4435
e-mail: artem.fomin@outlook.com

Gleb B. Sologub***
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0002-5657-4826
e-mail: glebsologub@ya.ru

Vladimir I. Vinogradov****
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0003-3773-9653
e-mail: vvinogradov@inbox.ru

The work is aimed at researching the possibility of using machine learning methods to build models for forecasting demand for new products in the online store Ozon.ru. Approaches to the solution that were not previously used in a specific task are proposed for consideration. Data on sales history and storage of goods at Ozon.ru are used as a sample. There is a description and analysis of the approximate loss of

For citation:
Osin A.A., Fomin A.K., Sologub G.B., Vinogradov V.I. Using Machine Learning Methods to Solve Problems of Forecasting Demand for New Products in the Internet Marketplace. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2020. Vol. 10, no. 4, pp. 41–50. DOI: https://doi.org/10.17759/mda.2020100404 (In Russ., abstr. in Engl.).

***Artem A. Osin,** Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000-0002-2664-1370, e-mail: artemosin1@yandex.ru

****Artem K. Fomin,** Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000-0003-3545-4435, e-mail: artem.fomin@outlook.com

*****Gleb B. Sologub,** PhD (Physics and Mathematics), Associate Professor, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000-0002-5657-4826, e-mail: glebsologub@ya.ru

******Vladimir I. Vinogradov,** PhD (Physics and Mathematics), Associate Professor, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000-0003-3773-9653, e-mail: vvinogradov@inbox.ru

the Ozon.ru website, the data used, the process of building a base model, and the results obtained. It describes the metrics used to evaluate the prediction results and makes a comparative analysis between the prediction results of the built model and the results of heuristically selected values.

Keywords: demand forecasting, new products, encoding, gradient busting, regression, preprocessing, data processing, machine learning.

References

1. Bisong E. Introduction to Scikit-learn // Building Machine Learning and Deep Learning Models on Google Cloud Platform 2019. P. 215–229.
2. Cerda P., Varoquaux G., Kégl B. Similarity encoding for learning with dirty categorical variables // Machine Learning. 2018. P. 1477–1494.
3. Ke G. et al. Lightgbm: A highly efficient gradient boosting decision tree // Advances in neural information processing systems. 2017. P. 3146–3154.
4. Redell N. Shapley Decomposition of R-Squared in Machine Learning Models // arXiv preprint arXiv:1908.09718. 2019.
5. Botchkarev, Alexei. “Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology.” // arXiv preprint arXiv:1809.03006. 2018.
6. Al Daoud E. Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset // International Journal of Computer and Information Engineering. 2019. P. 6–10.

Моделирование и анализ данных
2020. Том 10. № 4. С. 51–59
DOI: https://doi.org/10.17759/mda.2020100405
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2020. Vol 10, no. 4, pp. 51–59
DOI: https://doi.org/10.17759/mda.2020100405
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

УДК 004.85

Поиск контактной информации мошенников в объявлениях маркетплейсов

Смирнов Д.А.*
Московский авиационный институт
(национальный исследовательский университет)
(ФГБОУ ВО МАИ (НИУ)), г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0002-7092-2612
e-mail: daniil.smirnov2311@yandex.ru

Сологуб Г.Б.**
Московский авиационный институт
(национальный исследовательский университет)
(ФГБОУ ВО МАИ (НИУ)), г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0002-5657-4826
e-mail: glebsologub@ya.ru

В статье изложен подход к определению наличия в объявлении о продаже товаров или предоставлении услуг контактной информации: номера телефона, электронной почты, ссылки на сайт, id в социальных сетях.

Ключевые слова: работа с данными, построение моделей.

Для цитаты:
Смирнов Д.А., Сологуб Г.Б. Поиск контактной информации мошенников в объявлениях маркетплейсов // Моделирование и анализ данных. 2020. Том 10. № 4. С. 51–59. DOI: https:///doi.org/10.17759/mda.2020100405

***Смирнов Даниил Алексеевич,** студент, Московский авиационный институт (национальный исследовательский университет) (ФГБОУ ВО МАИ (НИУ)), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0002-7092-2612, e-mail: daniil.smirnov2311@yandex.ru

****Сологуб Глеб Борисович,** кандидат физико-математических наук, доцент кафедры, Московский авиационный институт (национальный исследовательский университет) (ФГБОУ ВО МАИ (НИУ)), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0002-5657-4826, e-mail:glebsologub@ya.ru

1. ВВЕДЕНИЕ

В настоящее время существует много различных интернет-сервисов для размещения объявлений о товарах, вакансиях и резюме на рынке труда, а также услугах от частных лиц и компаний. Такими сервисами могут воспользоваться как добропорядочные клиенты, так и злоумышленники, не по прямому назначению, а в целях обмана, мошенничества или спама. Представители этих сервисов заинтересованы в том, чтобы не допустить публикацию объявлений злоумышленниками, таким образом, возникает задача выявления таких объявлений, в том числе по их тексту.

В зависимости от конкретного сервиса принципы выявления объявлений злоумышленников могут быть различны. Так, в работе [1] описывается решение задачи выявления спама на сервисах типа Craigslist путем использования признаков на основе контента для обнаружения общего веб-спама и упоминается, что злоумышленники предпочитают не указывать номер телефона в объявлении. В то же время, на сервисах типа Avito, наоборот, правилами пользования ресурсом установлен запрет на указание любой контактной информации в любом месте объявления помимо специально выделенного окна сервиса [2], однако злоумышленники нарушают этот запрет и указывают номер телефона, маскируя его так, чтобы он не был автоматически распознан.

Так или иначе, наличие в тексте объявления контактной информации является важным признаком для выявления объявлений злоумышленников.

2. ПОСТАНОВКА ЗАДАЧИ

В качестве исходных данных составлены две таблицы, в которых каждая строка содержит информацию о конкретном объявлении: заголовок, описание, категория товара, подкатегория товара, цена, регион, город, дата публикации и маркер, указывающий на наличие и отсутствие контактной информации, нарушающей правила сервиса. Одна таблица используется для обучения модели, вторая – для валидации.

Требуется обучить модель, которая сможет для каждого объявления предсказывать, встретится ли там контактная информация. В результате работы обученная модель на вход получает объявление, а на выход выдает вероятность наличия контактной информации.

3. МЕТОД РЕШЕНИЯ

Для предсказания наличия контактной информации в объявлении нужно сформулировать алгоритм автоматических рекомендаций:

1. Проанализировать исходные данные, построить корреляционную матрицу для того, чтобы понять, как каждый столбец с информацией (заголовок, описание, категория и т.д.) влияет на таргетную переменную (признак, который может принимать значения 0 и 1, где 0 указывает на отсутствие контактной информации,

а 1 – на ее наличие), построить графики зависимости, где переменными будут выступать столбцы с информацией.

2. Провести поиск слов, которые влияют на таргетную переменную. На их основе составить новые столбцы с информацией об объявлении.
3. На полученном датасете подготовить обучающую и тестовую выборки.
4. Обучить предиктивные модели.
5. Оценить полученный результат.

Метрикой для оценки качества модели выбрана площадь (AUC – areaundercurve) под ROC-кривой (кривой ошибок). ROC-кривая – график, который позволяет оценить качество бинарной классификации. График показывает зависимость полноты (количество объявлений от общего числа реальных позитивных объектов, которые были предсказаны, как позитивный класс) от числа негативных объектов (объявления без контактной информации), которые предсказаны неверно.

Реализуем алгоритм на языке программирования Python. Для предобработки исходных данных хорошо подходят библиотеки pandas и numpy. Остальные шаги будем осуществлять при помощи библиотеки методов машинного обучения scikit-learn, библиотеки с реализацией градиентного бустинга lightgbm, морфологический анализатор pymorphy2, а также библиотеки для матчинга регулярных выражений re.

```
In [1]: from sklearn.linear_model import LogisticRegression
        from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
        from sklearn.ensemble import GradientBoostingClassifier
        from sklearn.feature_selection import RFECV, RFE
        import lightgbm as lgb
        import pandas as pd
        import numpy as np
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import LabelEncoder
        import re
        from sklearn.metrics import roc_auc_score
        import pymorphy2
        import phonenumbers
        from statistics import mean
```

Рис. 1. Импорт необходимых для работы библиотек

В частности, обработку исходных данных проведем с помощью библиотеки pandas. Считаем файл с помощью функции read_csv библиотеки pandas, после этого уберем строки, в которых есть незаполненные значения, с помощью функции dropna. Наконец, нужно переиндексировать строки, используя функцию reset_index. Обработка данных проиллюстрирована на рис. 2.

Для поиска контактной информации проводится поиск слов, которые указывают на это. Например, «тел. 89991234567» видно, что номер телефона указан после сокращения слова телефон «тел.». Значит, наличие «тел.» увеличивает вероятность нахождения контактной информации в объявлении. Если встречаем такое слово, то в информацию об объявлении добавляется значение 1, иначе 0. Реализация показана на рис. 3.

```
In [6]: df_train = pd.read_csv("train.csv")
        df_train.dropna(inplace=True)
        df_train.reset_index(drop=True, inplace=True)
        df_train.head(10)
```

Out[6]:

	title	description	subcategory	category	price	region	city	datetime_submitted	is_bad
0	Диван-кровать	Продаем диван-кровать. Удобный механизм - евро...	Мебель и интерьер	Для дома и дачи	7000.0	Россия	Москва	2019-06-01 00:00:15.180656	0
1	Кожух рулевой колонки Даф хф 91 4509834	Кожух рулевой колонки DAF XF 94 (60066004)/\n ...	Запчасти и аксессуары	Транспорт	2290.0	Россия	Москва	2019-06-01 00:00:44.317933	0
2	Дешёвый буст аккаунтов Dota 4	! Буст аккаунтов с ммр выше 1000ммр не беру ! ...	Предложение услуг	Услуги	200.0	Северная Осетия	Владикавказ	2019-06-01 00:00:50.249692	1
3	Телевизор sharp.Смарт тв.Интернет	Продам телевизор . Диагональ 450.наличие входа...	Аудио и видео	Бытовая электроника	25000.0	Калининградская область	Советск	2019-06-01 00:00:50.325799	1
4	Открытка-конверт	Открытки-конверты ручной работы/\nВыполнены в ...	Коллекционирование	Хобби и отдых	150.0	Ставропольский край	Ессентукская	2019-06-01 00:00:56.632655	0
5	Зимние шины Hankook Winter i*Pike RS6 W569	Размеры шин Hankook Winter i*Pike RS1 W319. Пр...	Запчасти и аксессуары	Транспорт	11000.0	Московская область	Железнодорожный	2019-06-01 00:01:03.471366	1
6	LADA Priora, 5046	приора 918 норма+кондиционер. 014 машина 16 в...	Автомобили	Транспорт	340000.0	Чеченская Республика	Грозный	2019-06-01 00:01:13.603386	1
7	Дверь входная	Продам дверь входную, данная дверь стояла от з...	Ремонт и строительство	Для дома и дачи	3000.0	Россия	Санкт-Петербург	2019-06-01 00:01:33.818452	0
8	Джинсы фирмы Gulliver	Продаю джинсы фирмы Gulliver. Отличная посадка.	Детская одежда и обувь	Личные вещи	500.0	Россия	Москва	2019-06-01 00:01:41.136540	0
9	Кроссовки nike AIR MAX 570 premium	Куплены на ASOS, не подошёл размер/\nСоответств...	Одежда, обувь, аксессуары	Личные вещи	8000.0	Россия	Москва	2019-06-01 00:01:55.506778	0

Рис. 2. Обработка исходных данных и результат

```
if desc.find("vk") != -1 or desc.find("id") != -1 or desc.find("вк") != -1 or \
    desc.find("вконтакте") != -1:
    line.append(1)
else:
    line.append(0)
```

Рис. 3. Пример обработки слов, влияющих на наличие информации о связи с продавцом в VK

Уже имеющиеся столбцы с информацией кодируем с помощью LabelEncoder(), который представляет текстовые признаки в виде чисел.

```
le.fit(df.category)
df_train['category'] = le.transform(df.category)
```

Рис. 4. Пример кодирования текстового столбца исходных данных

Таким образом, обогащаем наши данные информацией, которая сильно коррелирует с таргетной переменной. Данные выглядят следующим образом: первые столбцы – это преобразованные категориальные признаки с числовыми значениями, а после столбцы, которые могут принимать значения 1 и 0 – наличие или отсутствие эвристики, указывающей на наличие контактной информации.

В таком виде данные можно готовить к обучению модели. Разбиваем данные на 2 множества X и Y, X – информация, указывающая на наличие или отсутствие контактной информации, а множество Y – маркер присутствия контактной информации.

```
In [10]: df_train[range(len(df_train.iloc[0]) - 1)]
```

Out[10]:

	0	1	2	3	4	5	6	7	8	9	...	90	91	92	93	94	95	96	97	98	99
0	1.0	5.0	7000.0	2.0	4.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	3.0	3.0	2290.0	2.0	4.0	0.0	0.0	0.0	0.0	1.0	...	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
2	4.0	7.0	200.0	3.0	0.0	0.0	1.0	1.0	0.0	0.0	...	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
3	0.0	1.0	25000.0	0.0	6.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	5.0	4.0	150.0	4.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	3.0	3.0	11000.0	1.0	3.0	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
6	3.0	0.0	340000.0	5.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	1.0	8.0	3000.0	2.0	5.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	2.0	2.0	500.0	2.0	4.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	2.0	6.0	8000.0	2.0	4.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

10 rows × 100 columns

Рис. 5. Вид данных после обработки

```
In [11]: df_train = pd.DataFrame(data_for_train)
         x = np.array(df_train[[df_train.columns[i] for i in range(len(df_train.columns)-1)]])
         y = np.array(df_train[df_train.columns[len(df_train.columns)-1]])
```

Рис. 6. Разбиение данных

Дальше считываем тестовый датасет аналогично обучающему на рис. 2.

```
In [12]: df_test = pd.read_csv("val.csv")
         df_test.dropna(inplace=True)
         df_test.reset_index(drop=True, inplace=True)
```

Рис.7. Считывание и обработка тестовых данных

Следующим этапом является обучение модели на подготовленном датасете. Будем обучать такие модели, как логистическая регрессия, случайный лес для классификации и градиентный бустинг для классификации.

В основе логистической регрессии лежит логистическая функция [7]. Результат измеряется с помощью дихотомической переменной (в которой есть только два возможных результата), которой является целевая переменная Y. В поставленной задаче результатом является число, показывающее уверенность модели в наличии контактной информации в объявлении [6].

```
In [7]: from sklearn.linear_model import LogisticRegression

        classifier = LogisticRegression(random_state = 0)

        classifier.fit(xtrain, ytrain)
```

Рис. 8. Пример обучения логистической регрессии

Остаётся протестировать полученную модель. Выбранная ранее метрика оказалась равна лишь 0.77. (рис. 9)

```
In [11]: from sklearn.linear_model import LogisticRegression
         from sklearn.metrics import roc_auc_score

         classifier = LogisticRegression(random_state = 0)
         classifier.fit(xtrain, ytrain)
         y_pred = classifier.predict(xtest)


         print ("ROC-AUC: ", roc_auc_score(ytest, y_pred))

         /home/daniil/.local/lib/python3.6/site-packages/sklearn/utils/validation.py:72: DataConversionWarning: A column-vec
         tor y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using rav
         el().
           return f(**kwargs)
         ROC-AUC:  0.7691636803246837
```

Рис. 9. Построение модели логистической регрессии для поставленной задачи

Случайным лесом называется метод, подразумевающий создание нескольких деревьев принятия решений [7], которые затем принимают коллективное решение на основе голосования о том, каким образом классифицировать входящие значения [7].

Для построения модели случайного леса разделяем данные на обучающую и тестовую выборки и обучаем на подготовленных данных. После тестирование полученной модели получаем значение метрики ROC-AUC равное 0.82.

```
In [48]: model_forest = RandomForestClassifier()
         model_forest.fit(x,y)

Out[48]: RandomForestClassifier()

In [49]: print('roc-auc: {:.5f}'.format(roc_auc_score(y_test, model_forest.predict(x_test))))

         roc-auc: 0.81983
```

Рис. 10. Пример обучения и тестирования случайного леса

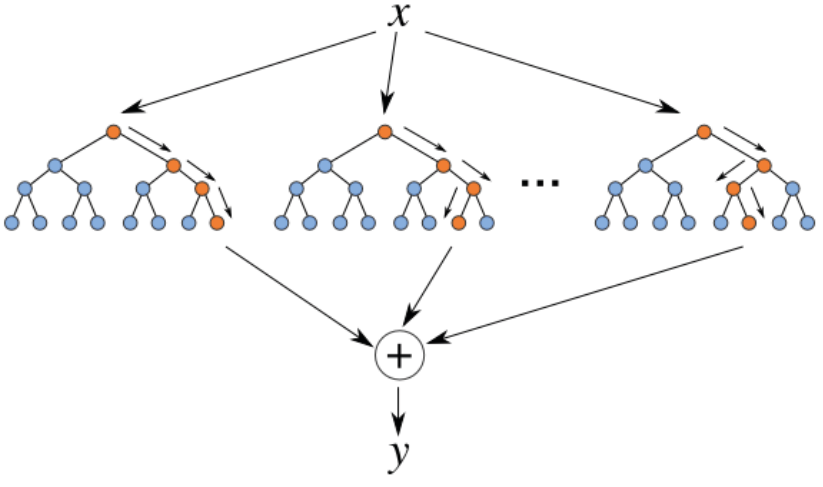

Рис. 11. Изображение случайного леса

Градиентный бустинг – это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей. Данный метод основан на градиентном спуске, основная идея которого заключается в том, чтобы идти в направлении наискорейшего спуска, которое задаётся антиградиентом:

$$\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \nabla F(\vec{x}^{[j]})$$

где $\lambda^{[j]}$ задает скорость градиентного спуска, $\vec{x}$ - аргумент функции $F(\vec{x}) : \mathrm{X} \to \mathbb{R}$, которую оптимизируем $F(\vec{x}) \to \min_{\vec{x} \in \mathrm{X}}$. Градиентом является вектор частных производных по каждой переменной.

Как средство борьбы с переобучением используем метод recursive feature elimination. Он последовательно перебором исключает признаки из набора и считает точность на тестовой выборке. Набор признаков с наилучшим результатом является выходом алгоритма.

```
In [8]: from sklearn.metrics import roc_auc_score

        predicted = model.predict(x_test)
        roc_auc = roc_auc_score(y_test, predicted)
        print('roc-auc: {:.2f}'.format(roc_auc))

        roc-auc: 0.96
```

Рис. 12. Точность модели градиентного бустинга на тестовых данных

Далее на рис. 13 представлен пример работы алгоритма: для трех случайных объявлений модель предсказывает вероятность наличия контактной информации.

```
In [58]: predicted = model.predict(x_test)
         for index, text in data_for_test.items():
             print()
             print('Описание объявления: \n{}\n'.format(text))
             print('Предсказание модели: {}'.format(predicted[index]))

         Описание объявления:
         Шины зимние б/у Marshal Assimetric I"Zen KW 61 681/41 r 63 диски AVA  литые универсальные 4 J 63 3/600 или 3/663,2
         ET 28~31. Забрать можно на Ярославском шоссе у МКАД или в г. Сергиев Посад. Могу подвезти(по договорённости) Состоя
         ние- на 6~5 сезона. БЕЗ ТОРГА и. 8 964 688 36 68

         Предсказание модели: 0.9920922326106677

         Описание объявления:
         Продам авто в хорошем состоянии.все вопросы по тел.Двигатель стоит родной,611л/с ,самый надёжный в этой линейке.Сал
         он фестфалия-трансформер,очень редкая комплектация:диван раскладывается и получаеться двухспальная кровать,выдвижно
         й столик,монитор,люк,электро жабры,электро стеклоподъемники,электро зеркала с подогревом,кожаный салон,передняя под
         ушка безопасности,потолок алькантара,кондиционер,вебаста с таймером,очень удобно зимой.ABS,ESP,ASR,блокировка перед
         них колёс (W),Теперь из не достатков:присутствует не большая коррозия в углу на правом пороге,притертость не значит
         ельная на сдвижной двери.От машины не избавляюсь,езжу каждый день,так что перекупам и стоянки просьба не беспокоит
         ь.Торг приветствуется,в подарок отдам зимнюю резину на литых дисках.звоните 8 девятьсот девятнадцать694 тридцать че
         тыре 29 .

         Предсказание модели: 0.9942272904703925

         Описание объявления:
         toyo observe g5-ice   без грыж и порезов звонить на( 89155903052)

         Предсказание модели: 0.9978764108431211
```

Рис. 13. Пример результата работы алгоритма

Также посчитаем метрику ROC-AUC по каждой категории объявлений для построенной модели. Реализация и результат продемонстрированы на рис. 14.

```
In [25]: ans_x = dict()
         ans_y = dict()
         cats = df_test["category"].unique()
         for cat in cats:
             ans_x[cat] = []
             ans_y[cat] = []
         for i in range(len(df_test.description)):
             ans_x[df_test.category[i]].append(x_test[i])
             ans_y[df_test.category[i]].append(y_test[i])
         auc_res = {}
         for cat in cats:
             data_x = ans_x[cat]
             data_y = ans_y[cat]
             pred = model.predict(data_x)
             roc_auc = roc_auc_score(data_y, pred)
             auc_res[cat] = roc_auc
         print(auc_res)
         mean(auc_res[i] for i in auc_res.keys())

         {'Транспорт': 0.9699697072971366, 'Для бизнеса': 0.7988688582537999, 'Для дома и дачи': 0.9318111268237161, 'Личные вещи': 0.80
         128781400516, 'Услуги': 0.8766427795312258, 'Бытовая электроника': 0.9437723925159236, 'Недвижимость': 0.9718166273316559, 'Хоб
         би и отдых': 0.9466941015089163, 'Животные': 0.9122730573710965, 'Работа': 0.8337569654902727}
```

Рис. 14. Алгоритм оценки модели и результат по каждой категории.

4. ЗАКЛЮЧЕНИЕ

Поставлена задача обучения модели поиска контактной информации мошенников в объявлениях маркетплейсов. Сформулирован алгоритм решения этой задачи на основе методов машинного обучения путем применения градиентного бустинга для классификации и отбора признаков. Описана реализация этого алгоритма на языке Python с применением свободно-распространяемых библиотек подпрограмм.

Предложенный алгоритм может быть использован для построения моделей поиска любой контактной информации человека в текстовых данных.

Литература

1. Spam Detection in Online Classified Advertisements / Hung Tran [и др.] // WebQuality 11: Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality / Association for Computing Machinery, 2011, pp.35–41, ISBN: 978–1-4503–0706–2, doi:10.1145/1964114.1964122
2. URL: https://m.avito.ru/info/polzovatelskoe_soglashenie
3. *Daniel Jurafsky, James H. Martin.* Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Second Edition. *Pearson Education International*, 2009. 1024 pp.
4. *Максимов Ю.А.* Алгоритмы линейного и дискретного программирования. – М.: МИФИ, 1980.
5. *Jones K.S.* A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation. – MCB University Press. – 2004. – Vol. 60, no. 5. – P. 493–502. – ISSN 0022–0418.
6. Электронный ресурс: URL:https://machinelearningmastery.com/logistic-regression-tutorial-for-machine-learning/
7. *Грас Джоэл.* DataScience. Наука о данных с нуля, [пер. с англ. Андрея Логунова]. – Санкт-Петербург, 2017. ISBN 978–5-9775–3758–2.

Search Contact Information of Scams in Classifieds of Marketplace

Daniil A. Smirnov*
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0002-7092-2612
e-mail: daniil.smirnov2311@yandex.ru

Gleb B. Sologub**
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0002-5657-4826
e-mail: glebsologub@ya.ru

The article describes an approach to determining whether an ad for sale or provision of services contains contact information: phone numbers, email addresses, website links, social media IDs etc.

Keywords: working with data, building models.

For citation:
Smirnov D.A., Sologub G.B. Search Contact Information of Scams in Classifieds of Marketplaces. *Modelirovanie i analiz dannykh = Modelling and Data Analysis,* 2020. Vol. 10, no. 4, pp. 51–59. DOI: https:///doi.org/10.17759/mda.2020100405 (In Russ., abstr. in Engl.).

References

1. Spam Detection in Online Classified Advertisements / Hung Tran [и др.]. WebQuality‘11: Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality / Association for Computing Machinery, 2011, pp.35–41, ISBN: 978–1-4503–0706–2, doi:10.1145/1964114.1964122
2. URL:https://m.avito.ru/info/polzovatelskoe_soglashenie
3. Daniel Jurafsky, James H. Martin. Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Second Edition. *Pearson Education International*, 2009. 1024 pp.
4. Maksimov Yu. A. Algoritmy lineinogo i diskretnogo programmirovaniya. – Moscow.: MEPHI, 1980.
5. Jones K.S. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation. *MCB University Press,* 2004. Vol. 60, no. 5, pp. 493–502. ISSN 0022–0418.
6. Elektronnii resurs: URL:https://machinelearningmastery.com/logistic-regression-tutorial-for-machine-learning/
7. Gras Dzhoel. Data Science. Nauka o dannykh s nulya, [per. s angl. Andreya Logunova]. – Sankt-Peterburg, 2017. ISBN 978–5-9775–3758–2.

***Daniil A. Smirnov,** undergraduate student of the Institute of Information Technology and Applied Mathematics, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000-0002-7092-2612, e-mail: daniil.smirnov2311@yandex.ru

***Gleb B. Sologub,** candidate of physical and mathematical sciences, associate professor of the Department of Mathematical Cybernetics, Institute of Information Technologies and Applied Mathematics, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000-0002-5657-4826, e-mail: glebsologub@ya.ru

Моделирование и анализ данных
2020. Том 10. № 4. С. 60–78
DOI: https://doi.org/10.17759/mda.2020100406
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2020. Vol 10, no. 4, pp. 60–78
DOI: https://doi.org/10.17759/mda.2020100406
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

◇◇◇◇◇◇◇ КОМПЛЕКСЫ ПРОГРАММ ◇◇◇◇◇◇◇

УДК 004.031.4

Организация клиент-серверного взаимодействия на одной электронно-вычислительной машине

Попков С.И.*
Московский государственный психолого-педагогический университет, г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0003-2566-1262
e-mail: rslw25@gmail.com

Проведено исследование имеющихся подходов к организации клиент-серверного взаимодействия. Особое внимание уделено практическому применению клиент-серверного взаимодействия на одной электронно-вычислительной машине. Проведен эксперимент с реализацией минимального кода сервера и применением виртуализации. Представлен результат эксперимента, описаны технические особенности и полученные результаты.

Ключевые слова: клиент, сервер, виртуализация, клиент-серверное взаимодействие.

Для цитаты:
Попков С.И. Организация клиент-серверного взаимодействия на одной электронно-вычислительной машине // Моделирование и анализ данных. 2020. Том 10. № 4. С. 60–78. DOI: https://doi.org/10.17759/mda.2020100406

1. ВВЕДЕНИЕ

Клиент-серверная архитектура распределенных информационных систем относится к одной из наиболее востребованных в настоящее время парадигм построения веб-приложений. Этому во многом способствует постоянное развитие сети Интернет и вклад в создание современных веб-технологий со стороны консорциума W3C [1] и других организаций.

Клиент-серверная архитектура предполагает наличие сервера – узла вычислительной системы с соответствующим программным обеспечением, позволяющим

Попков Сергей Игоревич, кандидат физико-математических наук, доцент факультета информационных технологий, заведующий лабораторией, Московский государственный психолого-педагогический университет, г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0003-2566-1262, e-mail: rslw25@gmail.com

обрабатывать запросы к серверу с целью получения актуализированной информации и ее дальнейшей передачи в доступном формате отправителю запроса – клиенту.

Однако, в силу различных обстоятельств, сеть Интернет может оказаться недоступной как физически (отсутствие соединения из-за аварии или качественных изменений естественных условий окружающей среды), так и логически (при работе с секретными или защищенными данными, когда компания или корпорация ограничивается локальной сетью). В последнем случае актуальной проблемой является поддержка работоспособности разработчика (или группы разработчиков) приложений в критических ситуациях, затрудняющих или делающих невозможным доступ к защищенной локальной сети без доступа к мировой сети Интернет (к частному случаю такой внештатной ситуации можно отнести удаленную работу в условиях глобальной пандемии).

В качестве одного из возможных решений для описанной ситуации можно предоставить в пользование разработчика общий снимок («snapshot») системы [2], очищенный от потенциальных уязвимостей и значимых для компании секретных данных. Однако, сам процесс такой очистки является достаточно трудоемким и содержащим потенциальные уязвимости ввиду участия человеческого фактора. Кроме того, такой подход не гарантирует совместимость вычислительных средств разработчика со снимком системы, поэтому обеспечение его работоспособности может привести к дополнительной трате ресурсов.

Другой подход предполагает абстрагирование от конкретных данных, где это возможно, и реализацию общей концепции работы системы с переносом клиент-серверной архитектуры. Это решение является более защищенным, но может оказаться неприемлемым на поздних стадиях разработки, где использование реальных рабочих данных может стать существенной необходимостью. Кроме того, этот способ требует затрат на перенос аппаратно-вычислительной среды, а также воссоздание ее работоспособности с соблюдением требований информационной безопасности.

Возможен и менее ресурсозатратный подход, если допустима атомарная автономная разработка, где участники проекта могут разрабатывать независимые узлы программного обеспечения, которые впоследствии легко соединяются между собой благодаря согласованным интерфейсам (например, микросервисная архитектура [3]). В этом случае допустима организация клиент-серверного взаимодействия на одной электронно-вычислительной машине. Цель данной работы – рассмотреть данную организацию и найти одно из возможных концептуальных решений для указанного случая.

2. КРАТКОЕ ОПИСАНИЕ ПРИНЦИПОВ РАБОТЫ КЛИЕНТ-СЕРВЕРНОГО ВЗАИМОДЕЙСТВИЯ БЕЗ РАЗДЕЛЕНИЯ АППАРАТНЫХ РЕСУРСОВ С ПРИМЕНЕНИЕМ ВИРТУАЛИЗАЦИИ

Для того, чтобы организовать работу клиента и сервера как взаимно независимых сущностей обычно требуется их аппаратное разделение. Действительно, для доступа

к серверу со стороны клиента используется одна программно-аппаратная среда, а для обеспечения работы сервера – другая. В связи с этим возникает вопрос, как в рамках поставленной задачи, где используется одна электронно-вычислительная машина, обеспечить разделение ресурсов? Ответом является практический прием, который позволяет изолировать логически взаимосвязанные вычислительные ресурсы, абстрагируясь от конкретной аппаратной реализации. Такой подход носит название «виртуализация».

Таким образом, на целевой машине, из основных аппаратных ресурсов, принадлежащих операционной системе, изначально установленной на машине (хостовая конфигурация, «host»), выделяется набор изолированных ресурсов, организованных таким образом, чтобы они представляли собой независимую виртуальную машину, на которую можно установить отдельную операционную систему (гостевая конфигурация, «guest»). Делается это при помощи специальных программных комплексов, включающих в себя низкоуровневые драйверы, прикладные программные и пользовательские интерфейсы. Эти комплексы носят название «гипервизоры», зачастую они написаны на нескольких языках программирования [4].

Существует множество платных и бесплатных гипервизоров, различающихся по характеристикам, поддерживаемым операционным системам, способам виртуализации, предоставляемым функциям. Среди них наиболее популярны такие программные комплексы, как Hyper-V [5], vSphere [6], VirtualBox [7], KVM [8] и др.

Подробный разбор различных программных комплексов, обеспечивающих виртуализацию, выходит за рамки данной работы. В данной работе будет использоваться VirtualBox как удобная для начинающего пользователя кросс-платформенная система с поддержкой полной аппаратной виртуализации и графическим интерфейсом. Это позволяет наглядно показать основные шаги по настройке клиент-серверного взаимодействия, но более искушенный пользователь, при необходимости, может легко перенести описанные шаги на другие средства виртуализации.

Как правило, клиентские машины используют операционную систему «Windows» ввиду ее высокой популярности и удобства для пользователя с любым уровнем опыта работы за компьютером [9]. Серверные машины, наоборот, зачастую используют «Linux» из-за большего количества удобных инструментов, более высокого уровня защищенности и надежности, производительности, выгоды с точки зрения цены и модифицируемости [10]. В данной работе мы также будем придерживаться аналогичной конфигурации.

3. УСТАНОВКА И НАСТРОЙКА ИНСТРУМЕНТОВ ДЛЯ ОБЕСПЕЧЕНИЯ РАБОТЫ ПРОЦЕССА ВИРТУАЛИЗАЦИИ

Прежде всего, необходимо установить средство виртуализации. Данный этап в работе подробно не рассматривается. Для гипервизора VirtualBox, который используется в данной работе, существует подробное описание шагов установки в руководстве пользователя [11] и подходящие зеркальные ссылки для загрузки дистрибутива

исходя из хостовой операционной системы и текущего сетевого региона, из которого осуществляется загрузка [12].

Далее рассматривается версия VirtualBox 6.1.16 с графическим интерфейсом на базе библиотеки Qt 5.6.2. Эта версия является наиболее актуальной [13] на момент написания статьи (в начале января 2021 года). В качестве операционных систем рассматривается 64-разрядная Windows версии 10.0.20279.1 [14] (host) и 32-разрядная Linux Mint LMDE 4 [15] (guest).

После установки гипервизора необходимо настроить элементы сетевого окружения, которые будут впоследствии осуществлять виртуальное клиент-серверное взаимодействие. Для этого необходимо открыть менеджер сетей хоста (комбинация клавиш «Ctrl+H» по умолчанию из главного окна менеджера VirtualBox). В появившемся окне создать сеть «VirtualBox Host-Only Ethernet Adapter» и настроить ее параметры вручную, задав адрес сети (например, исходя из умолчаний – частный IP-адрес 192.168.56.1), а маску сети оставить стандартной (255.255.255.0). IP-адрес будем использовать статический, поэтому DHCP-сервер можно отключить. Результат должен соответствовать изображению, представленному на рис. 1.

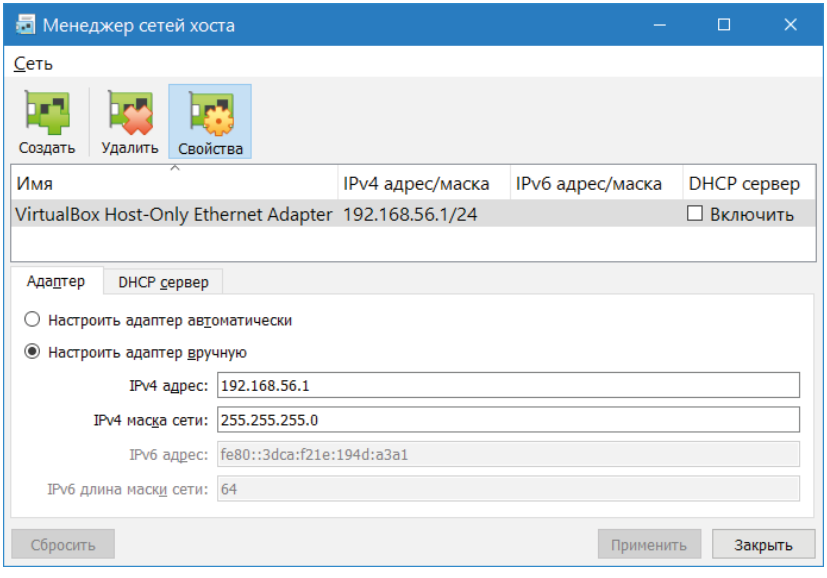

Рис. 1. Менеджер сетей хоста

Создание сети может потребовать предоставления прав системного администратора, поскольку эти изменения влияют на состояние хостовой операционной системы, в частности, на сетевые подключения. Открыв соответствующий раздел настроек операционной системы, необходимо убедиться в появлении нового подключения (рис. 2). Если подключение не обнаружено, может потребоваться осуществить перезапуск операционной системы.

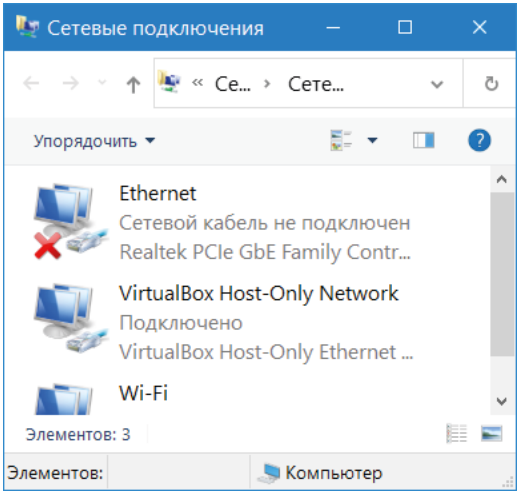

Рис. 2. Сетевые подключения. Сеть «VirtualBox Host-Only Network» отмечена в списке соединений как подключенная

Виртуальная сеть для взаимодействия между реальной хостовой и виртуальной гостевой машиной воспринимается как подключение по несуществующему кабелю. Сеть не подключена к сети Интернет и безопасна в использовании. На данный момент, однако, еще не создана и не подключена виртуальная машина, что видно из статистики после перезапуска адаптера в окне описания состояния соединения (рис. 3).

Перейдем к этапу создания виртуальной машины. Для начала необходимо определить тип и разрядность используемой операционной системы, которую планируется использовать в качестве гостевой. Можно просто указать общий тип (например, «Linux»), однако, делать этого не рекомендуется, поскольку тогда придется настраивать многие параметры самостоятельно. Удобнее предоставить средству виртуализации самостоятельно определить наиболее оптимальные параметры для работы гостевой операционной системы.

В случае с Linux Mint LMDE 4 известно, что эта операционная система основана на семействе дистрибутивов Debian [16]. Сам Debian является основой популярной операционной системы Ubuntu [17], что обеспечивает совместимость с большинством основных пакетов и средств, поддерживаемых дочерней операционной системой. При этом Debian – более легковесная операционная система и менее требовательная к ресурсам, что делает ее хорошим кандидатом на роль гостевой операционной системы в нашей работе для демонстрации базового примера клиент-серверного взаимодействия.

32-разрядная версия Linux Mint LMDE 4 менее требовательна к ресурсам, чем 64-разрядная, что позволяет ее использовать в фоновом режиме в роли сервера даже на старых аппаратных конфигурациях. Ограничения, накладываемые низкой разрядностью на взаимодействие с ресурсами, в рамках данной демонстрации не существенны.

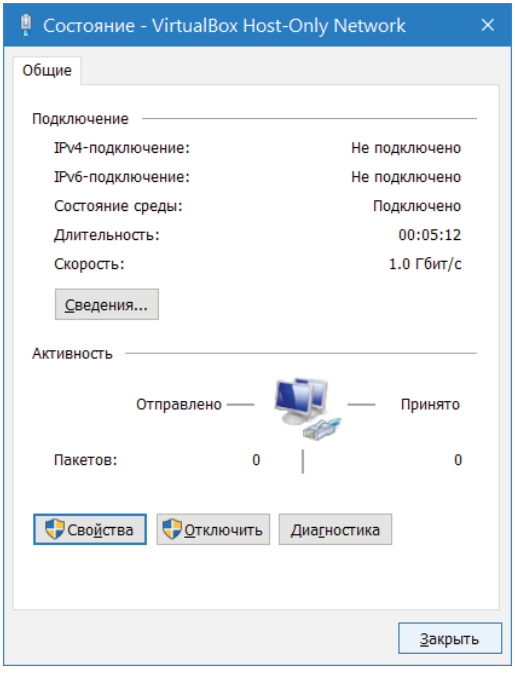

Рис. 3. Состояние сети

Создадим виртуальную машину (комбинация клавиш «Ctrl+N» по умолчанию из главного окна менеджера VirtualBox), назовем ее произвольным образом (например, «SelfServer»), укажем папку для хранения данных о виртуальной машине и заполним остальные поля согласно рис. 4.

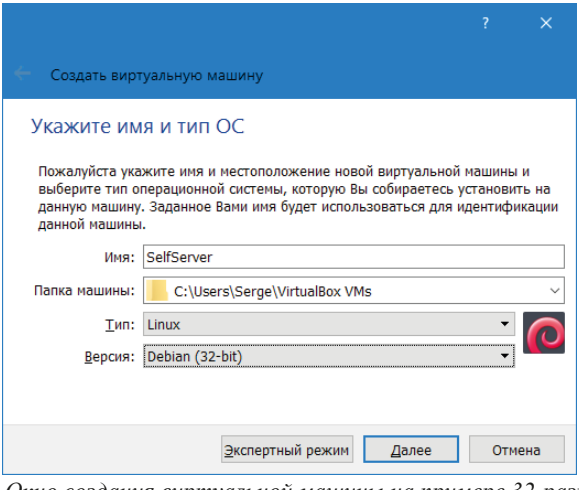

Рис. 4. Окно создания виртуальной машины на примере 32-разрядной операционной системы из семейства Debian

Объем памяти установим 1 ГБ (1024 МБ), виртуальный жесткий диск сделаем динамическим, размер установим 16 ГБ (чтобы гарантировать достаточное количество места для всех необходимых служб). Динамический жесткий диск будет работать медленнее фиксированного, однако, будет занимать меньше места (до тех пор, пока не будет полностью заполнен) и будет создан быстрее (для фиксированного виртуального жесткого диска необходимо сразу выделить весь запрошенный объем на физическом жестком диске).

После того, как все необходимые данные будут заполнены, а подготовка файлов для работы виртуальной машины будет завершена, в главном окне менеджера VirtualBox отобразится соответствующая виртуальная машина и краткое описание ее основных настроек с возможностью внесения изменений (рис. 5).

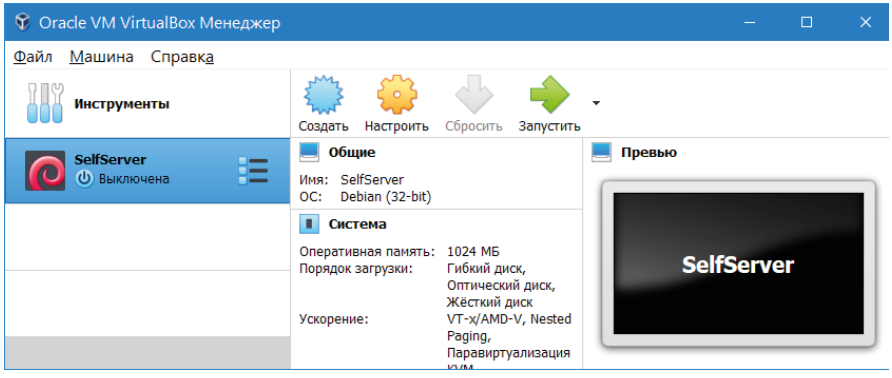

Рис. 5. Состояние созданной виртуальной машины

По умолчанию виртуальная машина подключена с помощью метода NAT («Network Address Translation», метод трансляции сетевых адресов) через виртуальный адаптер к той же сети, к которой подключена хостовая машина. Необходимо дополнительно подключить ранее созданную сеть для обеспечения клиент-серверного взаимодействия. Для этого в настройках виртуальной машины на вкладке «Сеть» устанавливаем переключатель «Включить сетевой адаптер» в установленное положение, выбираем тип подключения «Виртуальный адаптер хоста», имя ранее созданного и зарегистрированного в хостовой операционной системе виртуального сетевого адаптера «VirtualBox Host-Only Ethernet Adapter». Пример работающих настроек приведен на рис. 6.

По умолчанию общие папки не установлены. Если конфигурация виртуальной машины импортируется из внешнего источника, необходимо убедиться в отсутствии общих папок для обеспечения достоверности проводимых процедур: любое взаимодействие между клиентом и сервером должно осуществляться строго по выделенной для этого виртуальной сети – без доступа к сети Интернет и без использования дополнительных внешних механизмов, упрощающих решение задачи и нарушающих чистоту эксперимента.

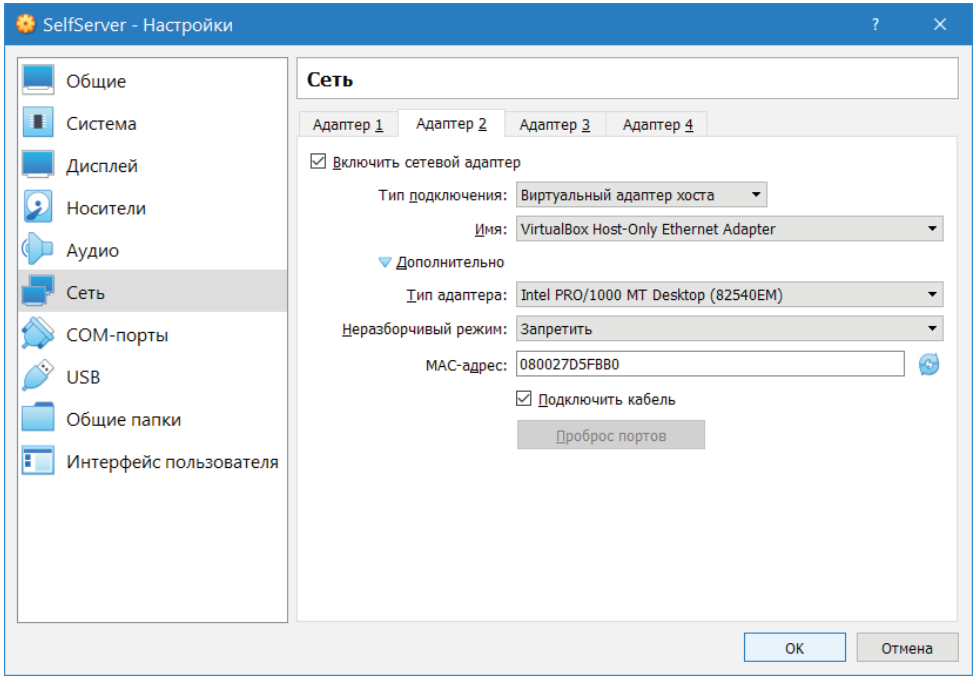

Рис. 6. Настройки сетевого адаптера

С домашней страницы сайта операционной системы Linux Mint LMDE 4 необходимо скачать официальный дистрибутив [18] в виде образа загрузочного диска в формате ISO [19], после чего запустить установщик и провести инсталляцию операционной системы на гостевую виртуальную машину. Более подробное описание процедуры выходит за рамки данной работы.

На данном этапе предполагается, что гостевая операционная система установлена и корректно работает на виртуальной машине, а сама машина успешно запускается. Иногда могут возникать проблемы с инициализацией виртуальной машины – как правило, они связаны с конфликтами между различными средствами виртуализации. В этом случае для корректного проведения описываемого эксперимента необходимо отключить или деинсталлировать другие средства виртуализации.

В некоторых случаях на хостовой операционной системе Windows 10 возможно возникновение ошибки средства виртуализации VirtualBox при запуске виртуальной машины. Код ошибки «VERR_INTNET_FLT_IF_NOT_FOUND» указывает на проблему с инициализацией и подключением виртуального сетевого адаптера «VirtualBox Host-Only Ethernet Adapter». Само окно выглядит как на рис. 7.

Эта ошибка является достаточно распространенной и решается переподключением виртуального сетевого адаптера на уровне хостовой операционной системы в окне состояния виртуального сетевого адаптера.

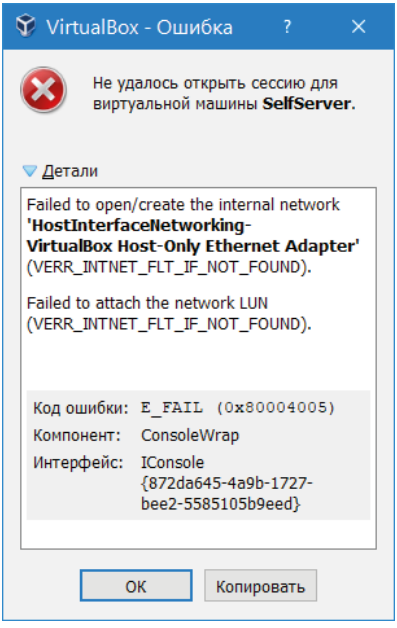

Рис. 7. Окно с сообщением об ошибке

В следующем разделе будет рассмотрен процесс настройки виртуальной машины.

4. ПРОЦЕСС НАСТРОЙКИ ВИРТУАЛЬНОЙ МАШИНЫ ДЛЯ ПОДДЕРЖКИ РАБОТЫ СЕРВЕРНЫХ СЛУЖБ И ОСУЩЕСТВЛЕНИЯ КЛИЕНТ-СЕРВЕРНОГО ВЗАИМОДЕЙСТВИЯ

После процедуры установки операционная система сообщает о проблемах подключения к сети хоста. Чтобы выявить проблему, можно проверить конфигурацию сети с помощью инструмента «ifconfig», как показано на рис. 8.

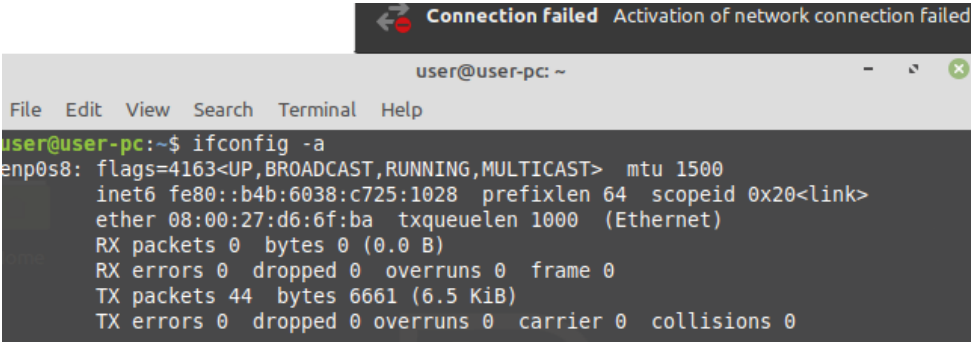

Рис. 8. Результат применения ifconfig после установки

Несмотря на то, что сеть на стороне виртуальной машины обнаруживается сразу, она остается недоступной для использования без конфигурации сетевого интерфейса и определения статического адреса для виртуальной машины, по которому она будет определяться в сети. В приводимом примере выбран адрес, согласующийся с адресом сети и маской, а именно – 192.168.56.10. При необходимости можно определить собственный адрес, исходя из конкретной конфигурации сети.

Для того, чтобы закрепить искомый адрес за гостевой виртуальной машиной, необходимо добавить в файл, отвечающий за регистрацию сетевых интерфейсов, соответствующие строки с командами. Эти команды будут инициализированы при следующем запуске операционной системы или инициализации сетевых интерфейсов. Пример внесенных изменений показан на рис. 9 (добавлены строки 4–7).

Повторный запуск инструмента «ifconfig» демонстрирует появление новой записи для конфигурации интерфейса сети хоста “inet 192.168.56.10 netmask 255.255.255.0 broadcast 192.168.56.255”. Теперь сетевой интерфейс настроен и позволяет обеспечить клиент-серверное взаимодействие. В качестве сервера выступает гостевая виртуальная машина с операционной системой на базе Linux (Debian), в качестве клиента – хостовая операционная система Windows 10, оснащенная современным браузером (в описываемом примере используется Google Chrome 87.0.4280.88).

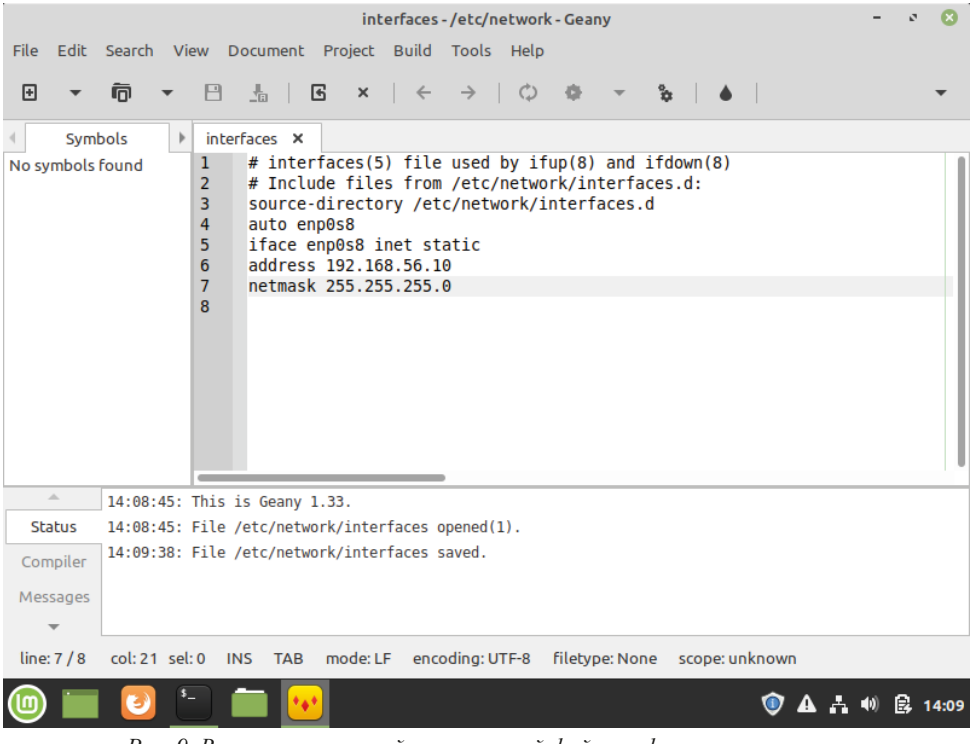

Рис. 9. Внесение изменений в системный файл конфигурации сетевых интерфейсов ОС Linux Mint LMDE 4 с использованием IDE Geany

Для того, чтобы убедиться, что сервер готов к выполнению своих функций, добавим для него роль SSH-сервера. SSH (“Secure Shell” [20], защищенная оболочка) – протокол для предоставления функций удаленного управления операционной системой. Позволяет безопасно передавать данные в незащищенной среде. Как правило, под соединение SSH выделяется порт TCP с номером 22, хотя при самостоятельной организации соединения рекомендуется менять стандартный номер порта во избежание bruteforce-атак со стороны потенциального взломщика [21]. Однако, в случае изолированной сети хоста для локально развернутой виртуальной машины подобными мерами безопасности можно пренебречь. Для проводимого в рамках данной работы эксперимента функционал SSH-сервера позволит убедиться в возможности полноценно взаимодействовать с сервером на стороне клиента, а также позволит подготовить необходимые файлы, описывающие сценарии взаимодействия с сервером.

Для установки SSH-сервера выполним в консоли команду “sudo apt install ssh”, после завершения установки проверить работу соответствующих служб можно, подключившись к серверу по локальной сети (через localhost командой “ssh localhost”). В этом случае виртуальная машина должна обратиться сама к себе и создать SSH-сессию, из которой можно выйти с помощью команды “logout” – в этом случае управление вернется в вызывающую консоль.

Для разрешения внешнего доступа следует добавить соответствующее правило во встроенный брандмауэр операционной системы, которое разрешит доступ к порту TCP извне. Для настройки правил доступа в Linux Mint можно воспользоваться как приложением с графическим интерфейсом, так и консолью. В проводимом эксперименте использовалась консольная команда “sudo iptables -A INPUT -p tcp –dport ssh -j ACCEPT”. Псевдоним «ssh» обозначает номер порта 22.

Для внешнего доступа к серверу потребуется клиент SSH, одним из наиболее популярных приложений в данной категории является PuTTY [22]. Данное приложение предоставляет различные настройки и позволяет выбрать требуемый протокол доступа, а также возможность работы с собственным форматом данных, не ограниченным конкретными спецификациями (режим “raw”).

Для проверки работы сервера SSH и возможности доступа к виртуальной машине не потребуется менять протокол и режимы работы. Достаточно убедиться, что виртуальная машина запущена, а в окне конфигурации подключения указать искомый адрес для подключения – 192.168.56.10.

Диалоговое окно настройки клиента SSH PuTTY с указанием необходимых данных о параметрах подключения перед получением доступа к виртуальной машине изображено на рис. 10.

Подключение осуществляется после нажатия кнопки “Open” в диалоговом окне. Открывается консольное окно, похожее на стандартный консольный интерфейс гостевой операционной системы. После авторизации (с указанием имени уполномоченного пользователя гостевой операционной системы и соответствующего пароля) в этом окне можно выполнить произвольные команды, для которых у соответствующей учетной записи имеется разрешение.

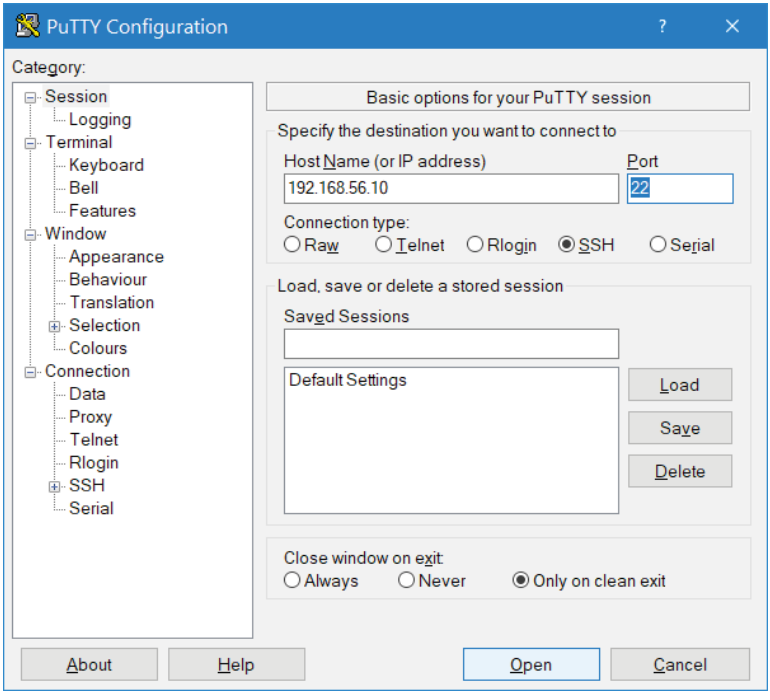

Рис. 10. Окно настройки SSH-клиента

Выполним команду создания каталога “mkdir” и вывода списка файлов на экран “ls” (рис. 11).

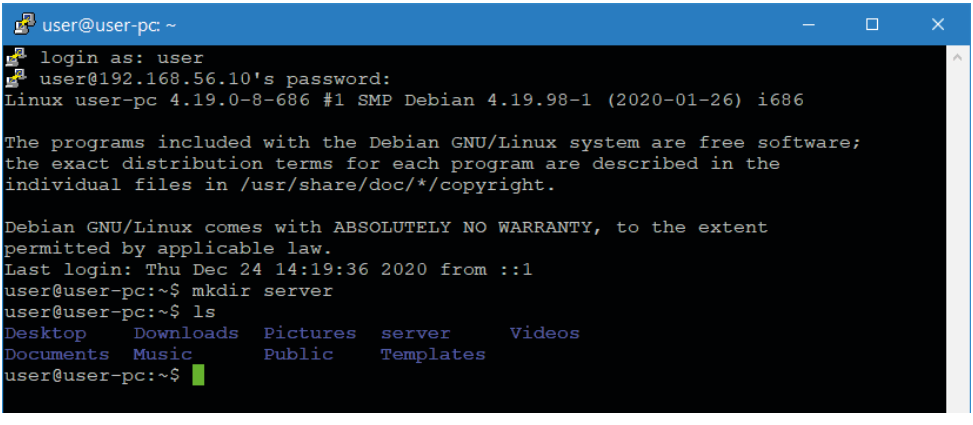

Рис. 11. Окно консоли PuTTY

В результате была создана папка “server”. Если открыть файловый менеджер в гостевой операционной системе, можно увидеть, что папка действительно была создана (рис. 12):

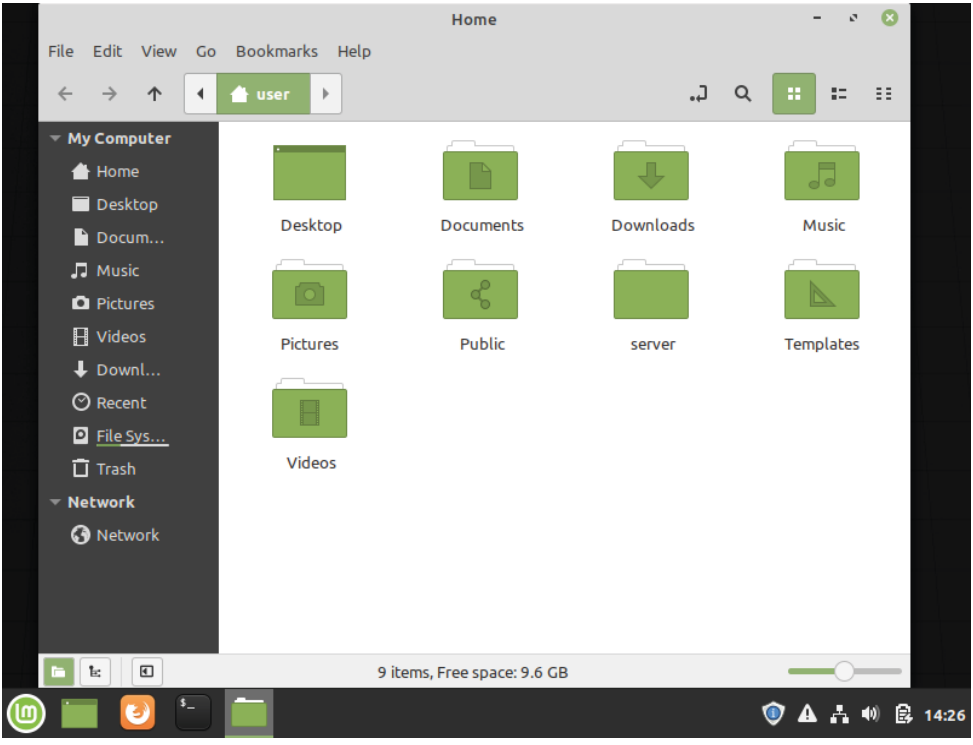

Рис. 12. Папка «server», созданная через подключение SSH, отображается в файловом менеджере гостевой операционной системы

Убедившись в том, что соединение настроено корректно, и сеть позволяет корректно осуществлять клиент-серверное взаимодействие, перейдем непосредственно к практической части эксперимента и создадим простейший сервер.

5. ПРОЦЕСС НАСТРОЙКИ ВИРТУАЛЬНОЙ МАШИНЫ ДЛЯ ПОДДЕРЖКИ РАБОТЫ СЕРВЕРНЫХ СЛУЖБ И ОСУЩЕСТВЛЕНИЯ КЛИЕНТ-СЕРВЕРНОГО ВЗАИМОДЕЙСТВИЯ

Исходный код для сервера (на базе фреймворка Flask [23]) показан на рис. 13.

Весь сценарий клиент-серверного взаимодействия, описанный исходным кодом в файле base.py, можно разбить на три части:

1) инициализация – загрузка необходимых для работы сервера библиотек, объектов и функций, запуск веб-приложения на основе выбранного фреймворка (строки 1–2);
2) создание веб-формы, отображаемой на стороне клиента и состоящей из кнопки отправки данных и поля ввода для сообщения со стороны пользователя (строки 4–11);

3) обработка результата, полученного со стороны клиента: передача клиенту информации о статусе обработки результата и сохранение полученного сообщения в файл на стороне сервера (строки 13–19).

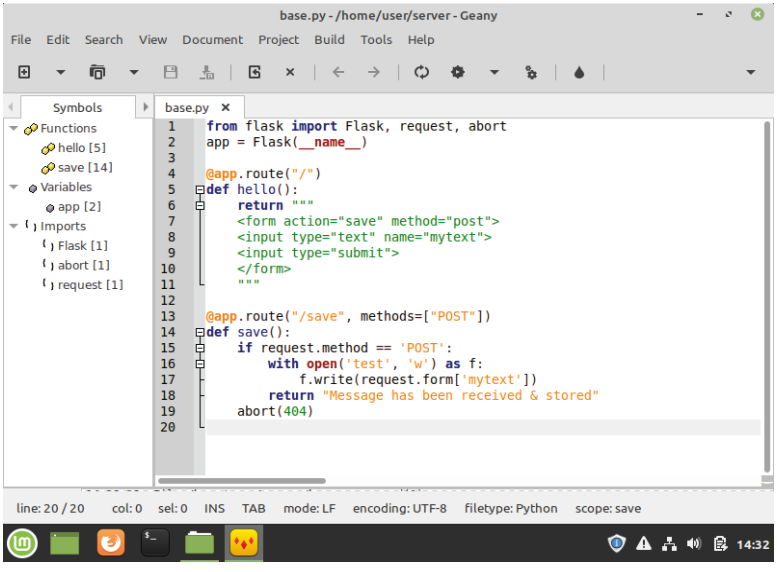

Рис. 13. Исходный код сервера

Для того, чтобы разрешить доступ к произвольному порту сервера со стороны клиента, необходимо задать соответствующие правила для брандмауэра. Это можно сделать через консольный интерфейс, воспользовавшись командами iptables и ufw. Для Linux Mint доступен графический интерфейс для добавления соответствующих правил (рис. 14).

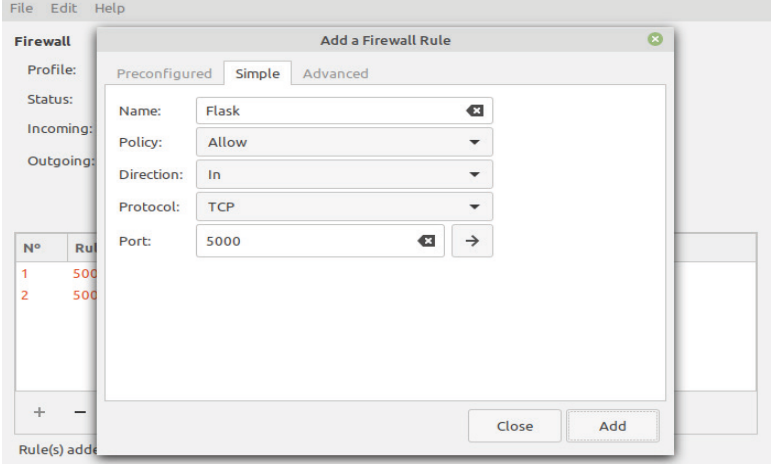

Рис. 14. Добавление правил брандмауэра через графический интерфейс Linux Mint

Установка фреймворка Flask и настройка брандмауэра выходит за рамки данной статьи; в качестве программной основы для сервера можно выбрать любое другое программное средство, настройки также будут специфичны для конкретных параметров конфигурации сервера.

В консоли сервера зададим переменную окружения FLASK_APP равной значению “base.py” (“export FLASK_APP=base.py”) и запустим приложение Flask из каталога, в котором хранятся файлы сервера, для всех доступных IP-адресов (“flask run –host=0.0.0.0”), после чего, убедившись, что в стандартном потоке вывода появилась строка «Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)», и сервер запущен корректно, на стороне клиента (хостовой операционной системы) необходимо открыть браузер и ввести адрес, привязанный к серверу, с портом веб-приложения сервера (по умолчанию для Flask это порт с номером 5000). Таким образом, адрес будет «http://192.168.56.10:5000». Результатом будет отображение минималистичной формы ввода данных, как показано на рис. 15.

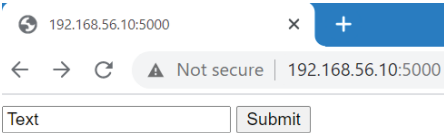

Рис. 15. Форма ввода данных

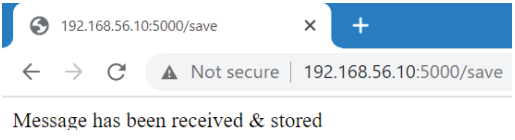

Рис. 16. Сообщение об успешной отправке данных

Введя в поле текст «Text» и нажав кнопку отправки на стороне клиента, пользователь инициирует на стороне сервера запуск обработчика и ответ, представленный на рис. 16.

На стороне сервера будет сгенерирован файл «test», его содержимым будет полученное сообщение (рис. 17). Это доказывает возможность клиент-серверного взаимодействия между хостовой и гостевой системами в качестве клиента и сервера, соответственно; клиент может как изменять состояние сервера (на примере создания файла с произвольным содержимым), так и получать данные от него (на примере получаемых сообщений в браузере).

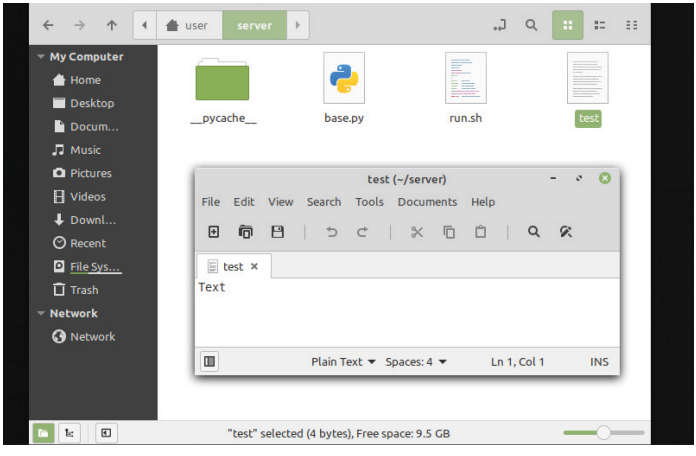

Рис. 17. Сгенерированный в ходе клиент-серверного взаимодействия файл

Продемонстрированный пример простейшего клиент-серверного взаимодействия легко можно расширить для более комплексных и полезных прикладных сценариев, например, фоновой автоматизации процессов сбора данных в сети, конвертации данных, а также процедуре отладки клиент-серверной архитектуры.

6. ЗАКЛЮЧЕНИЕ

В статье рассмотрена организация клиент-серверного взаимодействия на одной электронно-вычислительной машине, описан механизм виртуализации и продемонстрировано его использование с помощью одного из инструментов виртуализации, продемонстрирована настройка сети для осуществления клиент-серверного взаимодействия, показано практическое применение организации клиент-серверного взаимодействия на одной электронно-вычислительной машине в ходе эксперимента с запуском минимального кода сервера и обменом сообщениями между хостовой и гостевой системами; предложены идеи практически значимых расширений проведенного эксперимента для задач автоматизации процессов и отладки.

Литература

1. W3C – URL: https://www.w3.org/ (дата обращения 14.01.2020 г.)
2. Snapshot Red Hat Virtualization – URL: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.0/html/virtual_machine_management_guide/sect-snapshots (дата обращения 14.01.2020 г.)
3. Microservice Architecture – URL: https://microservices.io/ (дата обращения 14.01.2020 г.)
4. What is a hypervisor? – URL: https://www.vmware.com/topics/glossary/content/hypervisor (дата обращения 14.01.2020 г.)
5. Introduction to Hyper-V on Windows 10 – URL: https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/ (дата обращения 14.01.2020 г.)
6. What is vSphere? – URL: https://www.vmware.com/products/vsphere.html (дата обращения 14.01.2020 г.)
7. Oracle VM VirtualBox – URL: https://www.virtualbox.org/ (дата обращения 14.01.2020 г.)
8. Linux KVM – URL: https://www.linux-kvm.org/page/Main_Page (дата обращения 14.01.2020 г.)
9. Windows – URL: https://www.microsoft.com/en-us/windows (дата обращения 14.01.2020 г.)
10. Linux vs. Microsoft Windows Servers – URL: https://www.volico.com/linux-vs-microsoft-windows-servers/ (дата обращения 14.01.2020 г.)
11. Oracle® VM VirtualBox® User Manual – URL: https://www.virtualbox.org/manual/UserManual.html (дата обращения 14.01.2020 г.)
12. Download VirtualBox – URL: https://www.virtualbox.org/wiki/Downloads (дата обращения 14.01.2020 г.)
13. Changelog for VirtualBox 6.1 – URL: https://www.virtualbox.org/wiki/Changelog (дата обращения 14.01.2020 г.)
14. Announcing Windows 10 Insider Preview Build 20279 – URL: https://blogs.windows.com/windows-insider/2020/12/14/announcing-windows-10-insider-preview-build-20279/ (дата обращения 14.01.2020 г.)
15. Release Notes for LMDE 4 – URL: https://linuxmint.com/rel_debbie.php (дата обращения 14.01.2020 г.)

16. Debian – The Universal Operating System – URL: https://www.debian.org/ (дата обращения 14.01.2020 г.)
17. Ubuntu: Enterprise Open Source and Linux – URL: https://ubuntu.com/ (дата обращения 14.01.2020 г.)
18. Download LMDE 4 Debbie – URL: https://www.linuxmint.com/download_lmde.php (дата обращения 14.01.2020 г.)
19. Editions for Linux Mint 4 “Debbie” – URL: https://linuxmint.com/release.php?id=37 (дата обращения 14.01.2020 г.)
20. SSH (Secure Shell) – URL: https://www.ssh.com/ssh/ (дата обращения 14.01.2020 г.)
21. Смена порта SSH-сервера как мера защиты от брутфорса – URL: https://putty.org.ru/articles/change-default-sshd-port.html (дата обращения 14.01.2020 г.)
22. PuTTY: Telnet/SSH Клиент – URL: https://putty.org.ru/ (дата обращения 14.01.2020 г.)
23. Flask – URL: https://flask.palletsprojects.com/ (дата обращения 14.01.2020 г.)

Organization of Client-Server Interaction on a Single Computer

Sergei I. Popkov*
Moscow State University of Psychology and Education, Moscow, Russia
ORCID: https://orcid.org/0000-0003-2566-1262
e-mail: rslw25@gmail.com

A research of existing approaches to the organization of client-server interaction is conducted. Special attention is paid to the practical application of client-server interaction on a single computer. An experiment was conducted with the implementation of minimal server code and the application of virtualization. The result of the experiment is presented, the technical features and the results obtained are described.

Keywords: client, server, virtualization, client-server interaction.

For citation:
Popkov S.I. Organization of Client-Server Interaction on a Single Computer. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2020. Vol. 10, no.4, pp. 60–78. DOI: https://doi.org/10.17759/mda.2020100406 (In Russ., abstr. in Engl.).

References

1. W3C – URL: https://www.w3.org/ (req. date 14.01.2020 г.)
2. Snapshot Red Hat Virtualization – URL: https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.0/html/virtual_machine_management_guide/sect-snapshots (req. date 14.01.2020)
3. Microservice Architecture – URL: https://microservices.io/ (req. date 14.01.2020)
4. What is a hypervisor? – URL: https://www.vmware.com/topics/glossary/content/hypervisor (req. date 14.01.2020)
5. Introduction to Hyper-V on Windows 10 – URL: https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/ (req. date 14.01.2020)
6. What is vSphere? – URL: https://www.vmware.com/products/vsphere.html (req. date 14.01.2020)
7. Oracle VM VirtualBox – URL: https://www.virtualbox.org/ (req. date 14.01.2020)
8. Linux KVM – URL: https://www.linux-kvm.org/page/Main_Page (req. date 14.01.2020)
9. Windows – URL: https://www.microsoft.com/en-us/windows (req. date 14.01.2020)
10. Linux vs. Microsoft Windows Servers – URL: https://www.volico.com/linux-vs-microsoft-windows-servers/ (req. date 14.01.2020)
11. Oracle® VM VirtualBox® User Manual – URL: https://www.virtualbox.org/manual/UserManual.html (req. date 14.01.2020)
12. Download VirtualBox – URL: https://www.virtualbox.org/wiki/Downloads (req. date 14.01.2020)

***Sergei I. Popkov,** PhD in Physical and Mathematical Sciences, Assistant Professor of the faculty of information technologies, head of the laboratory, Moscow State University of Psychology and Education, Moscow, Russia, ORCID: https://orcid.org/0000-0003-2566-1262 , e-mail: rslw25@gmail.com

13. Changelog for VirtualBox 6.1 – URL: https://www.virtualbox.org/wiki/Changelog (req. date 14.01.2020)
14. Announcing Windows 10 Insider Preview Build 20279 – URL: https://blogs.windows.com/windows-insider/2020/12/14/announcing-windows-10-insider-preview-build-20279/ (req. date 14.01.2020)
15. Release Notes for LMDE 4 – URL: https://linuxmint.com/rel_debbie.php (req. date 14.01.2020)
16. Debian – The Universal Operating System – URL: https://www.debian.org/ (req. date 14.01.2020)
17. Ubuntu: Enterprise Open Source and Linux – URL: https://ubuntu.com/ (req. date 14.01.2020)
18. Download LMDE 4 Debbie – URL: https://www.linuxmint.com/download_lmde.php (req. date 14.01.2020)
19. Editions for Linux Mint 4 “Debbie” – URL: https://linuxmint.com/release.php?id=37 (req. date 14.01.2020)
20. SSH (Secure Shell) – URL: https://www.ssh.com/ssh/ (req. date 14.01.2020)
21. SSH-server port changing as an anti-bruteforce defensive measure – URL: https://putty.org.ru/articles/change-default-sshd-port.html (req. date 14.01.2020 г.)
22. PuTTY: Telnet/SSH Client – URL: https://putty.org.ru/ (req. date 14.01.2020)
23. Flask – URL: https://flask.palletsprojects.com/ (req. date 14.01.2020)

Моделирование и анализ данных
2020. Том 10. № 4. С. 79–87
DOI: https://doi.org/10.17759/mda.2020100407
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2020. Vol 10, no. 4, pp. 79–87
DOI: https://doi.org/10.17759/mda.2020100407
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

◇◇◇◇◇◇◇ МЕТОДЫ ОПТИМИЗАЦИИ ◇◇◇◇◇◇◇

УДК 51.7

Проблема адекватности метода анализа иерархий

Романчак В.М.*
Белорусский национальный технический университет,
г. Минск, Республика Беларусь
ORCID: https://orcid.org/ 0000-0001-9687-2919
e-mail: romanchak@bntu.by

Метод анализа иерархий (МАИ) является популярным методом решения многокритериальных задач. Многие исследователи подчеркивают простоту и естественность процедуры субъективного измерения. Но некоторые считают, что МАИ в целом является ошибочным и его нельзя применять на практике. Такой разброс мнений можно объяснить тем, что для МАИ не решена проблема адекватности. В данной работе предлагается модификация метода МАИ. Сформулирована математическая модель количественного измерения, которая содержит встроенный механизм проверки адекватности. Причем сохраняется способ измерения, а алгоритм расчета становится даже проще. Дело в том, что метод анализа иерархий основывается на предположении, что шкала отношений может быть получена посредством парного сравнения с использованием числовых суждений на основе абсолютной шкалы чисел. В качестве обоснования существования шкалы отношений рассматривается психофизический закон Фехнера. Но существует не один, а два психофизических закона. Существование двух психофизических законов является проблемой психофизики. Эта проблема может быть решена методом рейтинга. Чтобы преодолеть недостатки метода анализа иерархий также предлагается использовать метод рейтинга. В этом случае можно использовать фундаментальную шкалу метода МАИ. В качестве примера решается задача с использованием традиционной шкалы МАИ.

Ключевые слова: закон Фехнера, закон Стивенса, метод МАИ, метод анализа иерархий, рейтинг.

Для цитаты:
Романчак В.М. Проблема адекватности метода анализа иерархий // Моделирование и анализ данных. 2020. Том 10. № 4. С. 79–87. DOI: https://doi.org/10.17759/mda.2020100407

***Романчак Василий Михайлович,** кандидат физико-математических наук, доцент кафедры инженерной математики, Белорусский национальный технический университет, г. Минск, Республика Беларусь, ORCID: https://orcid.org/0000-0001-9687-2919, e-mail: romanchak@bntu.by

1. ВВЕДЕНИЕ

Метод МАИ [8], [9] привлекает внимание исследователей возможностью построить математическую модель принятия решений [2]. С другой стороны метод не имеет строгого математического обоснования. Поэтому существуют работы, в которых предлагаются различные модификации метода. Альтернативой методу МАИ могут выступать методы теории многофакторной полезности [10], специально созданные методики [3]. Даже сторонники метода согласны с тем, что метод МАИ не свободен от недостатков [11]. Существуют примеры некорректной работы метода МАИ [4]. Но метод МАИ по-прежнему остается популярным. Это связано с тем, что способ измерения в методе МАИ учитывает психологические особенности человека. Цель данной работы предложить модификацию метода МАИ. В этом случае сохраняется способ измерения, а алгоритм расчета становится даже проще. Возможность такой модификации объясняется тем, что фундаментальная числовая шкала метода МАИ соответствует уравнению психофизического закона Фехнера.

В настоящее время в психофизике существуют не один, а два психофизических закона, закон Фехнера и Стивенса. Наличие двух психофизических законов многими рассматривается как проблема. Решение проблемы возможно на основании метода рейтинга [5], [6]. В данной работе метод рейтинга предлагается использовать для получения результатов измерения совместно с методом МАИ. Сформулирована модель количественного измерения. Появляется возможность проверить адекватность результатов измерений и уменьшить объем экспериментальной работы. Причем для специалиста, знакомого с методом МАИ, все сводится к небольшим изменениям в схеме расчета. Для иллюстрации рассматривается пример оценки альтернатив, который ранее решался методом МАИ.

2. МОДЕЛЬ КОЛИЧЕСТВЕННОГО ИЗМЕРЕНИЯ

Для измеряемых объектов $\omega_1, \omega_2, \ldots, \omega_n$ сформулируем определение количественного измерения. Считаем, что результат измерения имеет двойственную природу. С одной стороны, результат измерения получают экспериментальным сравнением пары объектов (ω_i, ω_j) по величины. С другой стороны, р*езультат измерения* равен разности значений: $u_i - u_j$, или отношению значений: v_i / v_j. Поэтому можно говорить о двух *способах* количественного измерения.

Пример 1. Пусть значения массы объектов $\omega_1, \omega_2, \ldots, \omega_6$ изменяются так, что отношение значений массы двух последовательных объектов равно двум: $v_{i+1} / v_i = 2$, v_i – значение массы объекта, $i = 1, 2, \ldots, 5$. Результат измерения первым способом находим по формуле $u_i - u_j = i - j$. Для измерения вторым способом используем заданные в условии отношения. Тогда получаем выражения $v_i / v_j = 2^i / 2^j$, где $i, j = 1, 2, \ldots, 6$. На основании результатов измерения значения u_i можно определить с точностью до произвольной аддитивной постоянной, а значения v_i – с точностью до произвольной мультипликативной постоянной.

Для частных значений величины в табл. 1 существует отображение $u_i = \ln(v_i) / \lambda$, где $\lambda = \ln(2)$, которое устанавливает взаимно однозначное соответствие между значениями. Кроме того, отображение преобразует отношение значений v_i / v_j в разность значений $u_i - u_j$, $i, j = 1, 2, \ldots, 6$:

$$\lambda (u_i - u_j) = \ln(v_i / v_j). \quad (1)$$

Таблица 1

Значения, полученные двумя способами

$\boldsymbol{u_i}$	1	2	3	4	5	6
$\mathbf{v}_i$	2	2^2	2^3	2^4	2^5	2^6

Отображение $u = \ln(v) / 1$ является изоморфизмом множества положительных действительных чисел с операцией деление на множество действительных чисел с операцией вычитание. С точки зрения алгебры изоморфные структуры можно не различать, они эквивалентны [1]. Равенство (3) означает, что изоморфизм преобразует одни результаты измерения в другие. Такие результаты измерения будем называть эквивалентными.

Результаты измерения рассматриваемые в Примере 1 эквивалентны, так как выполняется равенство (1). Из примера следует, что измерения можно производить как субъективными, так и объективными методам. Зная отношение значений: v_i / v_j, можно рассчитать разность значений: $u_i - u_j$, и наоборот.

Отношению значений: u_i / u_j, и разности значений: $v_i - v_j$, можно придать физический смысл. Но отношения значений: u_i / u_j, и разности значений $v_i - v_j$, лишены физического смысла и в математической модели рейтинга не определены. Закону Фехнера соответствует разности значений [7]. Поэтому возникают сомнения в целесообразности использовать в методе МАИ отношения значений.

Определение. Равенство (1) заменим двумя выражениями

$$R_{ij} = \lambda(u_i - u_j), \quad (2)$$

$$R_{ij} = \ln(v_i / v_j), \quad (3)$$

где $i, j = 1, 2, \ldots, n$, λ – постоянная масштаба. Отображения (4) и (5) будем называть *рейтингом*.

В зависимости от способа измерения рейтинг можно определить по формулам (2) или (3). Значения рейтинга не зависят от способа измерения. Это означает, что рейтинг есть величина, которая инвариантна к способу измерения. Подчеркнем, что определение рейтинга не сводится к замене переменных, а опирается на такое фундаментальное понятие алгебры как изоморфизм. Экспериментальным подтверждением существования двух способов измерения (4) и (5) являются психофизические законы Фехнера и Стивенса [7]. Рейтинг можно использовать для определения функции $u(x)$ на множестве X.

Пусть для отображения $R(x, x_0)$ существует функция $u(x)$ для которой выполняется равенство

$$u(x) - u(x_0) = \lambda R(x, x_0), \quad (4)$$

для всех $x \in X$, $x_0 \in X$, λ – постоянная, $\lambda > 0$. Тогда отображение $R(x, x_0)$ также будем называть рейтингом. Определение (4) обобщается на случай произвольного количества аргументов. Пусть существует функция $u(x, y)$, $x \in X$, $y \in Y$, для которой выполняется обобщенное равенство (4). Зафиксируем произвольную точку (x_0, y_0).

Определение. Множества X и Y взаимно независимы относительно результатов измерения, если выполняются равенства

$$u(x, y) - u(x_0, y) = \lambda R_1(x), \quad (5)$$

$$u(x, y) - u(x, y_0) = \lambda R_2(y), \quad (6)$$

здесь $R_1(x), R_2(x)$ – частные значения рейтинга. Тогда из равенств (5) и (6) следует аддитивное представление функции

$$u(x, y) = \lambda(R_1(x) + R_2(y)) + u(x_0, y_0) \quad (7)$$

λ – постоянная масштаба. Аналогичным образом можно получить мультипликативное представление функции, если использовать определение рейтинга (3) и сформулировать определение независимости для отношения значений функции, используя формулы, аналогичные (5), (6).

3. ПРИМЕР КОЛИЧЕСТВЕННОГО ИЗМЕРЕНИЯ

Проведем анализа данных модифицированным методом МАИ, используя пример из монографии [8]. Измерения, полученные методом МАИ, соответствуют закону Фехнера. Поэтому значения, полученные в методе МАИ будем вычитать, а не делить.

Пример. Существует три альтернативных способа распределение электроэнергии между тремя потребителями C_1, C_2, C_2. Кроме того выявлено три основных фактора (критерия), влияющих на благоприятное социальное и политическое положение: x – экономика, y – экология, y – безопасность. Матрица парных сравнений метода МАИ [8] представлена в таблице 2.

Таблица 2

Матрица парных сравнений

x / y	***x***	***y***	***z***
x	1	5	3
y	1/5	1	3/5
z	1/3	5/3	1

Заменим в этой матрице отношения на разности. Например, если в матрице парных сравнений табл. 1 стоит число 3, заменим его разностью 3–1. Аналогично, если стоит отношение 1/5 – заменим его на разность 1–5. В итоге получим результаты измерения (таблица 3).

Таблица 3

Результаты измерения (разности значений)

U	x	y	z
x	0	4	2
y	-4	0	-2
z	-2	2	0

Пусть значения функции $U(x, y, z)$ являются численной характеристикой состояния государства. Считаем, что факторы x, y, z взаимно независимы. По аналогии с представление (7) выберем аддитивную модель значений величины в виде

$$U(x, y, z) - U(0, 0, 0) = k_1x + k_2y + k_3z \quad (8)$$

где k_1, k_2, k_3 – постоянные коэффициенты, x, y, z – переменные (факторы), $0 \le x \le 1$, $0 \le y \le 1$, $0 \le z \le 1$. Вектор (x, y, z) численно характеризует экономику (x), экологию (y) и безопасность (z) . Вектор $(0, 0, 0)$ соответствует наименее благоприятному, а вектор $(1, 1, 1)$ – наиболее благоприятному состоянию экономики.

Теперь можно преобразовать некоторые вербальные оценки в математические формулы. Например, естественно определить влияние первого фактора как разность значений $U(1, 0, 0) - U(0, 0, 0)$. Совместное влияния первого и второго фактора находим по формуле $U(1, 1, 0) - U(0, 0, 0)$. Выражение $U(1, 0, 0) - U(0, 1, 0)$ характеризует на сколько первый фактор превосходит второй. Далее с помощью таблицы 3 находим коэффициенты модели (8).

Из таблицы 3 следует, что влияние первого фактора превосходит влияние второго на 4 единицы. Следовательно, выполняется уравнение $k_1 - k_2 = 4$. Аналогично получаем равенство $k_3 - k_2 = 2$. Все остальные уравнения, которые можно получить из таблицы 3, являются следствие этих двух уравнений. Столбцы таблицы 3 соответствуют различным планам эксперимента. Следовательно результаты расчетов, выполненные по разным планам эксперимента, совпадают. Такое совпадение вряд ли является случайным. Поэтому считаем, что результаты измерения адекватны. Для нахождения неизвестных коэффициентов k_1, k_2, k_3 данных таблицы 3 недостаточно. Необходима дополнительная информация. Например, эксперт может считать, что совместное влияния первого и второго фактора является сильным. Тогда с помощью фундаментальной шкалы МАИ [8] получим недостающее третье уравнение $k_2 + k_1 = 6$. В этом случае значения коэффициентов $k_1 = 5$, $k_2 = 1$, $k_3 = 3$. Так как вектор коэффициентов определен с точностью до постоянной масштаба, то удобно перейти к нормализованному вектору (0,56; 0,11; 0,33). У нормализованного вектора сумма коэффициентов равняется единица. Для сравнения вектор приоритетов МАИ в данной задаче имеет вид (0,65; 0,13; 0,22).

Определяем зависимость фактора x от способа распределения C_i. Это означает, что необходимо найти численные значения $x_i = x(C_i)$. Тогда разность значений $x_i - x_j$ характеризует в какой мере экономика с распределением C_i, превосходит экономику

с распределением C_j. Если преобразовать матрицу парных сравнений из рассматриваемого примера [8], получим таблицу 4.

Таблица 4

Результаты измерений

$x_i - x_j$	C_1	C_2	C_3
C_1	0	2	4
C_2	-2	0	1
C_3	-4	-1	0

Чтобы исключить грубые ошибки удобно использовать стандартные статистические методы. Коэффициенты корреляции $\rho(C_1, C_2) = 0{,}982$, $\rho(C_1, C_3) = 0{,}961$, $\rho(C_2, C_3) = 0{,}996$ незначимы. Найдем вектор средних значений C, используя только второй и третий столбец таблицы 4. Коэффициент корреляции для этих двух столбцов наибольший. Столбцы C_2, C_3 таблицы 5 получены из столбцов таблицы 4 путем вычитания наименьшего элемента каждого столбца.

Таблица 5

Оценка влияния на экономику

C_2	C_3	C	x
3	4	3,5	1
1	1	1	0,286
0	0	0	0

Коэффициенты корреляции $\rho(C, C_2) = 0{,}999$, $\rho(C, C_1) = 0{,}999$ близки к единицы и значимы. Поэтому считаем, что вектор средних значений C соответствует результатам измерения. Нормализованные значения вектора x находятся в последнем столбце таблицы. Если теперь найти средние значения C, используя три столбца таблицы 5, то значимым будет только коэффициент $\rho(C, C_2)$. Поэтому делаем вывод, что первый столбец содержит грубые ошибки, а второй и третий столбец адекватны количественной модели измерений. Аналогично определяем значения $y_i = y(C_i)$ и $z_i = z(C_i)$, $y = (1; 0{,}818; 0)$ и $z = (1; 0{,}5; 0)$. Теперь, используя линейную модель (8), оцениваем влияние альтернатив C_1, C_2, C_3. Если выбрать значение $U(0, 0, 0) = 0$, то выполняются равенства $U(1, 1, 1) = 1$, $U(x_2, y_2, z_2) = 0{,}42$. Оценки альтернатив определены с точностью до линейного преобразования. Чтобы сопоставить с методом МАИ в таблице 6 оценки приведены к общему интервалу [0,12; 0,62].

Таблица 6

Оценка альтернатив

	C_1	C_2	C_3
Теория рейтингов	0,62	0,33	0,12
Метод МАИ	0,62	0,26	0,12

Из анализа таблицы 6 следует, что разница между оценками альтернатив C_2 и C_3 меньше, чем между оценками альтернатив C_1 и C_2. Причем, в отличии от метода МАИ, у нас есть основания так считать.

4. ЗАКЛЮЧЕНИЕ

В тех случаях, когда необходим простой способ оценки альтернатив, можно комбинировать метод МАИ и метод рейтинга. Метод рейтинга, в отличие от метода МАИ, является аксиоматическим. Для проверки соответствия результатов измерения модели рейтинга можно применять различные критерия. Из разобранного примера следует, что для количественной оценки альтернатив, достаточно частично определить матрицу парных сравнений. Предлагаемый алгоритм не сложнее метода МАИ.

Литература

1. *Курош А. Г.* Лекции по общей алгебре. М.: Физматлит, 1973. 400 с.
2. *Осипова В.А., Дубинина К.С.* Применение алгоритмов теории графов к упрощенному методу анализа иерархий // Моделирование и анализ данных. 2019. Том 9. № 3. С. 24–31.
3. *Подиновский В.В*. Количественная важность критериев // Автоматика и телемеханика. 2000. № 5. С. 110–123.
4. *Подиновский В.В., Подиновская О.В*. О некорректности метода иерархий // Проблемы управления. 2011. № 1. С. 8–13.
5. *Романчак В.М.* Измерение нефизической величины // Системный анализ и прикладная информатика. 2017. № 4. С. 39–44.
6. *Романчак В.М. С*убъективное оценивание вероятности // Информатика. 2018. Том.15. № 2. С. 74–82.
7. *Романчак В.М.* Субъективные измерения (теория рейтингов). // Журнал Белорусского государственного университета. Философия. Психология. 2020. № 3. С. 87–98.
8. *Саати Т.Л.* Принятие решений. Метод анализа иерархий / Т.Л. Саати. Москва: Радио и связь, 1989. 316 с.
9. *Саати Т.Л.* Относительное измерение и его обобщение в принятии решений. Почему парные сравнения являются ключевыми в математике для измерения неосязаемых факторов // Журнал «Cloud Of Science». 2016. Т. 3. № 2. [Электронный ресурс] URL: https:// cloudofscience.ru/sites/ default/ files/ pdf/CoS_3_171.pdf. (дата обращения: 17.12.2019).
10. *Dyer J.S.* Remarks on the analytic hierarchy process // Management science. 1990. V. 36. № 6. Pp. 249–258.
11. *Whitaker R.* Criticisms of the analytic hierarchy process: why they often make no sense // Mathematical and Computer Modelling. 2007. Vol. 46. P. 948–961.

The Problem of Adequacy of the Analytic Hierarchy Process

Vasily M. Romanchuk*
Belarusian national technical University, Minsk, Belarus
ORCID: https://orcid.org/ 0000-0001-9687-2919
e-mail: romanchak@bntu.by

The Analytic hierarchy process (AHP) is a popular method for solving multi-criteria problems. However, the problem of the adequacy of the AHP method is not solved. Opponents of the Analytic hierarchy process believe that the AHP as a whole is erroneous and cannot be applied in practice. Proponents of the method believe that the disadvantages of the method are compensated by a simple measurement procedure. In this paper, a modification of the AHP method is proposed. A mathematical model of measurement is formulated, which contains a built-in mechanism for checking adequacy. moreover, the measurement method is preserved, and the calculation algorithm becomes even simpler. The fact is that the Analytic hierarchy process is based on the assumption that the scale of relations can be obtained by pairwise comparison using numerical judgments based on the absolute scale of numbers. Fechner's psychophysical law is considered as a justification for the existence of the scale of relations. But there are not one, but two psychophysical laws. The existence of two psychophysical laws is a problem of psychophysics. This problem can be solved by the rating method. To overcome the disadvantages of the Analytic hierarchy process, it is also proposed to use the rating method. The use of the rating method makes it possible to use the fundamental scale of the AHP method. As an example, the problem is solved using the traditional AHP scale.

Keywords: Fechner's law, Stevens' law, MAI method, hierarchy analysis method, rating.

For citation:
Romanchuk V.M. The Problem of Adequacy of the Analytic Hierarchy Process. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2020. Vol. 10, no.4, pp. 79–87. DOI: https://doi.org/10.17759/mda.2020100407 (In Russ., abstr. in Engl.).

References

1. Kurosh, A.G. Lekcii po obshhej algebre. [Lectures on General algebra], Moscow: Fizmatlit, 1973. 400 p.
2. Osipova, V.A., Dubinina K.S. Application of graph theory algorithms to the simplified method of hierarchy analysis// Modeling and data analysis. 2019. Vol. 9, no 3. P. 24–31.
3. Podinovskii, V.V. Quantitative Importance of Criteria // Automation and Remote Control. 2000. Vol. 61, no. 5. Part 2. – P. 817–828.

***Vasily M. Romanchuk,** PhD in Phys. – Matt., Associate professor, Chair of "Engineering mathematics", Belarusian national technical University, Minsk, Belarus, ORCID: https://orcid.org/0000-0001-9687-2919, e-mail: romanchak@bntu.by

4. Podinovski V.V., Podinovskaya O.V. O nekorrektnosti metoda analiza ierarkhiy. *Problemy upravleniya,* 2011, No.1, pp. 8–13.
5. Romancak, V.M. (2017). Measurement of non-physical quantity. System analysis and applied Informatics, 4, 39–44. (In Russ., abstr. in Engl.).
6. Romancak V.M. (2018). Subjective estimation of probability. Informatics, 15(2), 74–82. (In Russ., abstr. in Engl.).
7. Romanchak, V.M. (2020). Subjective measurements (rating theory). Journal of the Belarusian State University. Philosophy and Psychology, 3, 87–98.
8. Saaty, Th. L. Prinyatie resheniy. Metod analiza ierarkhiy [Decision making. Analytic hierarchy process]. Moscow, Radio and communication, 1989. 316 p
9. Saaty, Th. L. Relative Measurement and Its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors – The Analytic Hierarchy/Network Process. URL: http://www.rac.es/ficheros/doc/00576.PDF (accessed: 17.12.2020).
10. Dyer, J.S. Remarks on the analytic hierarchy process // Management science. 1990. V. 36. № 6. Pp. 249–258.
11. Whitaker, R. Criticisms of the analytic hierarchy process: why they often make no sense // Mathematical and Computer Modelling. 2007. Vol. 46. P. 948–961.